TRANSIMS Version 1.1

Volume Four

Calibrations, Scenarios, and Tutorials

LA-UR-00-1766

(updated 20 August 2001)

Disclaimer

These archived, draft documents describe TRANSIMS, Version 1.1, covered by the university research license. However, note that the documentation may be incomplete in some areas because of the ongoing TRANSIMS development. More recent documentation (for example, Version 2.0) may provide additional updated descriptions for Version 1.1, but also covers code changes beyond Version 1.1.

1. CALIBRATION

1.1 Introduction

The TRANSIMS calibration suite is used to understand fundamental traffic flow characteristics of the TRANSIMS microsimulation. In order to determine the effects of driving rules, the calibration suite provides controlled tests of traffic flow behavior. The test networks are simplified situations where elements of the microsimulation can be tested in isolation.

1.2 Calibration Networks

The calibration networks provide the following test cases for evaluation of driving logic:

- One-lane freeway traffic to test if car-following behavior generates reasonable fundamental diagrams.
- Three-lane freeway traffic to test if passing-lane-changing behavior generates reasonable fundamental diagrams. Lane usage can also be evaluated.
- Merging at stop and yield signs and left turns against opposing traffic to test for acceptable flow rates at non-signalized intersections.
- Signalized intersection to evaluate flow rates and to test lane-changing behavior for plan following.

The freeway network (one and three lanes) is a 1000-cell (75 km) circle where vehicles enter on one side of the circle (cell 1), and flow and density measurements are taken on the opposite side of the circle (cells 491 – 495). The vehicle density is continuously increased from 0 to 0.5 (0 veh/km/lane to 66 veh/km/lane) during the simulation run.

To test merge behavior at stop and yield signs at an unsignalized intersection, an incoming link with one lane is added to the circle at cell 500. The incoming vehicles are removed at site 750.

To test left turns against opposing traffic, two links are added to the circle. An opposing link that ends at cell 500 on the circle provides vehicles that will make a left turn across the traffic on the circle. The vehicles turn across the circle onto a link that begins at cell 500 where they eventually exit from the simulation.

The signalized intersection includes a three-lane approach link. Three one-lane links exit from the intersection: one left, one straight, and one right. Incoming vehicles attempt lane changes on the approach link in order to follow their intended movement at the intersection. The intersection has a signal with a 60-second green phase and a 60-second red phase.

Note: For more information, refer to *TRANSIMS traffic flow characteristics*, by K. Nagel, P. Stretz, M. Pieck, S. Leckey, R. Donnelly, and C. L. Barrett. The paper was presented to the Transportation Research Board in Washington, D.C., in January 1998.]

1.3 Circle Network

1.3.1 Overview

The Circle Network, with merge and turn lanes, is used to calibrate and test the microsimulation. There are two circle networks—a one-lane and a three-lane circle. Multiple calibrations are run on these basic networks. First, freeway traffic is calibrated by looking at traffic—uninterrupted by turns or merges—around the one- and three-lane circles. The one-lane circle demonstrates car following behavior, while the dynamics of lane changing is apparent using the three-lane circle. The speed limit on the circle is 37.5 meters per second (five cells per second) for the Freeway calibrations and 22.5 meters per second (three cells per second) for both the merge and left-turn calibration networks.

Both the one- and three-lane circles have intersections (or nodes) where vehicles can merge into the traffic moving around the circle or cross the traffic on the circle. These intersections are used to measure (and calibrate) the effects of gap acceptance parameters for left turns and merging.

1.3.2 Description

Both the one- and three-lane circle calibration networks have thirteen unidirectional links. Links 1 through 7 form the circle. These links and the order of their connections are shown in Fig. 1. The link numbers are given in the rectangles surrounding the diagram. Traffic on the circle moves from link 1 to 2 to 3 etc., before returning to link 1 from link 7. As shown in Fig. 1, the actual roadway is laid out as a square, but traffic behaves as if it were a circle. All nodes on this portion of the network act as *seamless* nodes. Vehicles do not pause or slow down at the corners of the box. The maximum speed on these seven links is 37.5 meters per second (five cells per second).

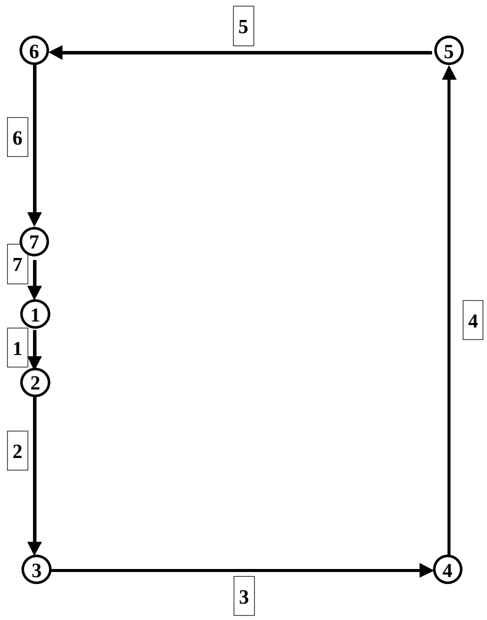


Fig. 1. The structure and connections of the seven links and nodes making up the one- and three-lane circles.

Links 3, 4 and 5 are 1875 meters long. The shorter links 2 and 6 are 750 meters long, while the shortest links 1 and 7 are but 187.5 meters. Links 8 through 12 allow for traffic to merge into the traffic on link 1 or cross the traffic at the node joining links 1 and 7, node 1. Fig. 2 shows these links.

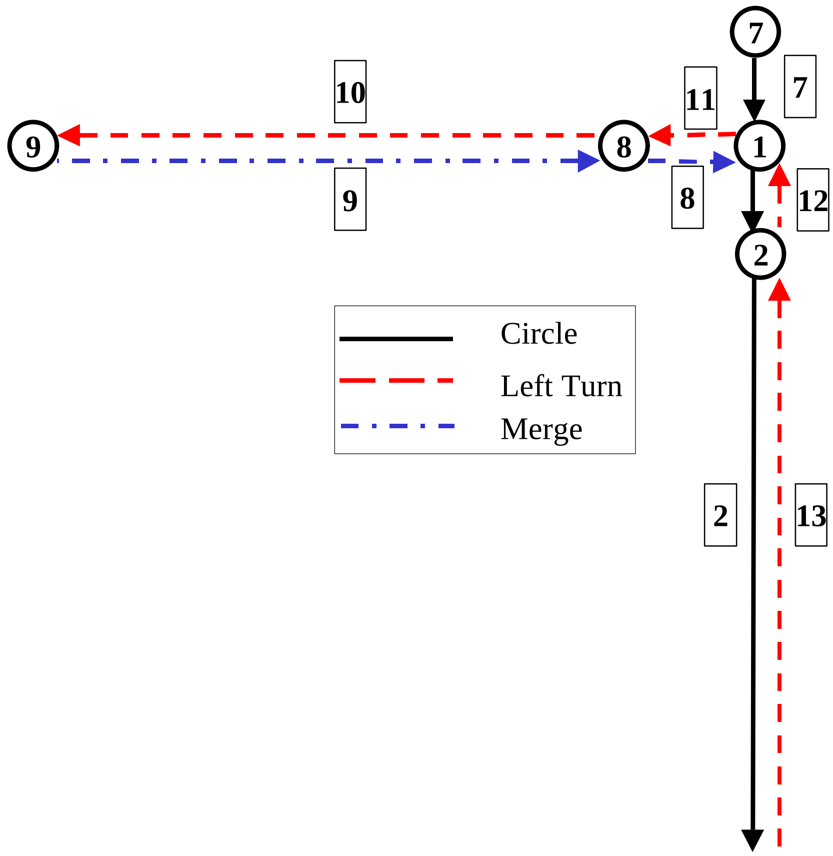


Fig. 2. Links of the circles, left turn, and merge networks are shown. Links 1, 2, and 7 are on the circle. Link 12 allows for left turns across the traffic at node 1. Link 8 permits merges into the traffic below link 7 on link 1.

For right-turn or merge calibrations, traffic originates on link 9, moves to link 8, and merges into the circle traffic on link 1. Stop, yield, or no controls may be placed at the node to control traffic moving from link 8 to link 1.

To calibrate gap acceptance for left turns across traffic, traffic starts on link 13, moves to link 12, crosses the circle traffic at the node, and continues on to links 11 and 10. Vehicles turning left must yield to the circle traffic. The circle traffic is not affected by the turning movements of the other vehicles.

1.3.3 Usage

The Circle Networks are used to calibrate the Traffic Microsimulator. Each is invoked using a preset plan file, and the output is filtered to produce an analysis file containing the statistics of interest.

1.3.3.1 Calibration Filters

The filters on the freeway, merge, and left-turn calibration runs collect summary data on five-cell sample blocks placed at specified locations on the calibration networks (Table

1). The data is summarized over three-minute intervals. The sample blocks cut across all lanes of the link, but data is reported on a lane-by-lane basis, as well as the link totals.

Table 1. Calibration Network – Sampling Locations

Network	Sample Block Location	Sampled Type
Freeway	Sites 491 – 495 on the circle	Circle vehicles
Merge	Sites 491 – 495 (Link 7) on the circle Sites 501 – 505 (Link 1) on the circle	Circle vehicles Merging vehicles
Left Turn	Sites 491 – 495 (Link 7) on the circle Sites 1 – 5 (Link 11) on exiting link	Circle vehicles Left-turn vehicles

1.3.3.2 Freeway Calibrations

One- and three-lane freeway traffic is calibrated by moving traffic around the circle on links 1 to 7. Configuration files for these two calibrations are in files *$TRANSIMS_HOME/data/calibration/freeway1/freeway1.cfg* and *$TRANSIMS_HOME/data/calibration/freeway3/freeway3.cfg*. The plan files contain vehicle plans that start vehicles on one of the seven links and move the vehicles around the circle. The density of vehicles is continuously increased by adding more vehicles to the network. The vehicles continue around the circle and are not removed. A small snippet of one vehicle's plans is given below. In this plan, vehicle and individual number 1 start at parking location 4 on link 4 then pass through nodes 5, 6, 7, 1, 2, 3 and 4 in order.

```
1 0 1 1 1 1
2 4 2 1 2
3360358 3360360 1
1 0 1
3361
1 0
5 6 7 1 2 3 4
5 6 7 1 2 3 4
5 6 7 1 2 3 4
5 6 7 1 2 3 4
```

The microsimulation for the two freeway calibrations is carried out using the following commands:

```
      % cd $TRANSIMS_HOME/data/calibration/freeway1
      % csh
$TRANSIMS_HOME/data/calibration/freeway1/scripts/run_calib.csh
      % cd $TRANSIMS_HOME/data/calibration/freeway3
      % csh
$TRANSIMS_HOME/data/calibration/freeway3/scripts/run_calib.csh
```

Snapshot output is collected on link 7. This output is filtered using the program *$TRANSIMS_HOME/bin/FreewayFilter***.** This filter is invoked by the shell scripts above using the following command (where <> denotes input data):

```
% $TRANSIMS_HOME/bin/FreewayFilter <# lanes> <snapshot file name> <output file>
```

This produces output files with the format in Table 2.

Table 2. Freeway filter format.

Field	Interpretation
Simulation Time	Seconds since simulation start.
Lane	Lane number.
Density	Vehicles/km/lane at the sample block on the circle.
Flow	Vehicles/hour/lane at the sample block on the circle.

The columns contain the vehicle flows and roadway densities on the lanes on link 7. These can be plotted to produce a diagram showing the relationship between the flows and the densities.

1.3.3.3 Merge Calibrations

The configuration file for the merge calibration simulation is *$TRANSIMS_HOME/data/calibration/merge2/merge2.cfg***.** The plan file for this simulation contains plans for vehicles traveling around the circle as in the freeway calibrations. Additional vehicles are given plans that start at parking location 13 on link 8 and merge into the traffic on link 1 at node 1. These vehicles proceed through nodes 2 and 3. They are removed from the network at the parking location 3 on link 3. A sample of these plans is given below.

```
1000001 0 1 1 1 1
0 13 2 3 2
8753 8753 1
1 0 1
5
1000001 0
1 2 3
```

The microsimulation for the merge calibration is carried out using the following command lines:

```
    % cd $TRANSIMS_HOME/data/calibration/merge2

    % csh
$TRANSIMS_HOME/data/calibration/merge2/scripts/run_calib.csh
```

Output is collected on links 1 and 7 and is filtered for analysis using the program *$TRANSIMS_HOME/bin/MergeFilter***.** The file produced by this filter contains the vehicle flow and density on link 7 and the flow of the merging vehicles on link 1. These data may be plotted to assess the merge rate as a function of the flow of oncoming traffic. The format of this file is shown in Table 3.

Table 3. Merge filter format.

Field	Interpretation
`Simulation Time`	Seconds since simulation start.
`Lane`	Lane number.
`Flow-7`	Flow of circle traffic in vehicles/hour/lane (link 7).
`Density-7`	Density of circle traffic in vehicles/km/lane (link 7).
`Flow-1`	Flow of merging vehicles in vehicles/hour/lane (link 1).

This filter is invoked by the script using the following command line:

```
    % $TRANSIMS_HOME/bin/MergeFilter <# lanes> <snapshot file
name> <output file>
```

1.3.3.4 Left-Turn Calibrations

The number of vehicles making a left-hand turn across traffic is calibrated using the left-turn plan set. The configuration file for this simulation is found in *$TRANSIMS_HOME/data/calibration/left2/left2.cfg*. Here, a baseline set of vehicles traverses the circle as in the freeway or circle calibration studies. Meanwhile, vehicles turning left start at parking location 16 on link 12, cross oncoming traffic at node 1, and move onto link 11. They proceed through node 8 to link 10 and are removed from the simulation at parking location 9. The plan of *left-turn* vehicle number 1000001 is given below.

```
1000001 0 1 1 1 1
0 16 2 9 2
8628 8628 1
1 0 1
4
1000001 0
1 8
```

This microsimulation calibration is invoked with the following commands:

```
    % cd $TRANSIMS_HOME/data/calibration/left2

    % csh
$TRANSIMS_HOME/data/calibration/left2/scripts/run_calib.csh
```

An output file is produced by filtering the vehicle snapshot data from links 7 and 11. The filtering code is in *$TRANSIMS_HOME/bin/LeftturnFilter*. It is invoked by the script using the following command:

```
    % $TRANSIMS_HOME/bin/LeftturnFilter <# lanes> <snapshot
file name> <output file>
```

This filter computes the flow of the vehicles that make the left turn to link 11 and the flow and density of the vehicles on link 7. The format of this file is shown in Table 4.

Table 4. Left-turn filter format.

Field	Interpretation
`Simulation Time`	Seconds since simulation start.
`Lane`	Lane number.
`Flow-7`	Flow of circle traffic in vehicles/hour/lane (link 7).
`Density-7`	Density of circle traffic in vehicles/km/lane (link 7).
`Flow-11`	Flow of left-turn vehicles in vehicles/hour/lane (link 11).

1.4 Tee Network

1.4.1 Overview

Lane changing behavior and plan following are studied using a network in the form of a "T." Vehicles start at the bottom of the "T" in three lanes and try to move to the correct lane to continue straight ahead, turn right, or make a left turn. As with the circle networks, this network is designed to test the dynamics of the Traffic Microsimulator rather than the complete TRANSIMS framework. Vehicle plan sets for traffic on this network are produced *offline*—not using the Route Planner.

The intersection at the top of the "T" is controlled by a traffic signal. Therefore, this network is also used to evaluate vehicle behavior at the signal.

1.4.2 Description

The Tee Network has five links and is shown in Fig. 3. Links 1 and 2 are three-lane freeway type links with maximum speeds of 30 meters per second (four cells per second).

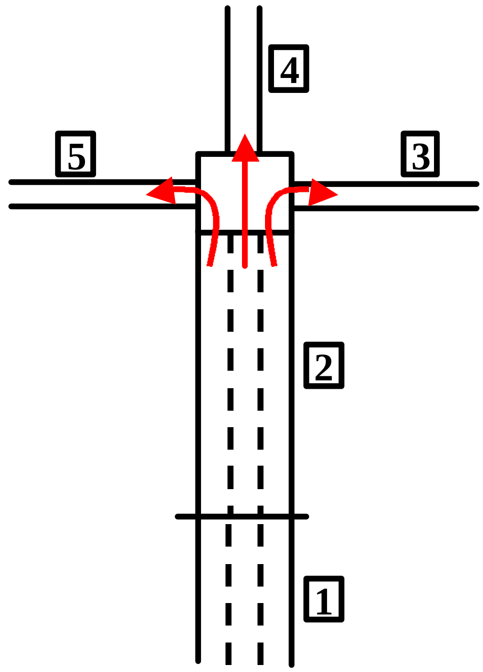


Fig. 3. The five links of the Tee Network are shown. Vehicles enter on link one and exit on one of the links 3, 4, or 5.

In this network, the left lane of link 2 is connected to the single lane of link 5. The middle lane is connected to link 4, and the right lane to link 3. The intersection connecting links 2 through 5 is controlled by a signal operating under various phasing cycles. The set of plans for this network has vehicles starting midway up link 1 and exiting on one of the links 3 through 5.

1.4.3 Usage

The tee calibration tests the weaving behavior of vehicles as they approach the intersection at node 2. A script to exercise this network is found in *$TRANSIMS_HOME/data/calibration/tee/scripts/run_calib.csh***.** All vehicles in this plan file start at parking location 1 on link 1 and proceed through node 1 to link 2. At node 2, one-third of the vehicles, those numbered 1 to 3600, turn right to link 3; one-third, numbered 3601 to 7200, move straight ahead to link 4; and the one-third with vehicle numbers greater than 7200 turn left to link 5. The vehicles are placed on link 1 in a random lane and attempt to enter the correct lane on link 2. Three of these plans, one for each of the turning movements, follow:

```
1 0 1 1 1 1
1 1 2 3 2
1 1 1
1 0 1
4
1 0
1 2
```

```
3601 0 1 1 1 1
1 1 2 4 2
1 1 1
1 0 1
4
3601 0
1 2

7201 0 1 1 1 1
1 1 2 5 2
1 1 1
1 0 1
4
7201 0
1 2
```

The Traffic Microsimulator is invoked using the configuration file *TRANSIMS_HOME/data/calibration/tee/tee.cfg*. The following commands are used to run the simulation.

```
      % $TRANSIMS_HOME/data/calibration/tee
      % csh
$TRANSIMS_HOME/data/calibration/tee/scripts/run_calib.csh
```

Correct microsimulation behavior is determined by assessing the vehicle snapshot data for link 2. Additionally, event data will indicate any vehicle that is unable to move to the correct lane for movement through the intersection. This information is obtained by filtering the raw data. The filtering code is in *$TRANSIMS_HOME/bin/TeeFilter*. It is invoked with the following command:

```
% $TRANSIMS_HOME/bin/TeeFilter <light cycle> <snapshot file name> <signal evolution file>
```

The light cycle time for all calibration networks given here is 60 seconds. The output from the filters is in two files: *tee.lane_state.60* and *tee.vehicle_turns.60*. The format of these files is shown in Table 5 and Table 6.

Table 5. Signalized intersection filter – vehicle lane state.

Field	Interpretation
`Time`	Seconds since simulation start.
`Box Distance From Node 2`	Starting distance of box in meters measured from the node from which the vehicles are traveling away .
`Light Color`	State of traffic control (`g`=green, `r`=red).
`Light Cycle`	Length of traffic control cycle in seconds.
`#in-1-to-1`	Number of vehicles in the sample box during the 30-second interval that were in lane 1 and planned to turn left at the intersection.
`#in-1-to-2`	Number of vehicles in the sample box during the 30-second interval that were in lane 1 and planned to go straight at the intersection.

Field	Interpretation
`#in-1-to-3`	Number of vehicles in the sample box during the 30-second interval that were in lane 1 and planned to turn right at the intersection.
`#in-2-to-1`	Number of vehicles in the sample box during the 30-second interval that were in lane 2 and planned to turn left at the intersection.
`#in-2-to-2`	Number of vehicles in the sample box during the 30-second interval that were in lane 2 and planned to go straight at the intersection.
`#in-2-to-3`	Number of vehicles in the sample box during the 30-second interval that were in lane 2 and planned to turn right at the intersection.
`#in-3-to-1`	Number of vehicles in the sample box during the 30-second interval that were in lane 3 and planned to turn left at the intersection.
`#in-3-to-2`	Number of vehicles in the sample box during the 30-second interval that were in lane 3 and planned to go straight at the intersection.
`#in-3-to-3`	Number of vehicles in the sample box during the 30-second interval that were in lane 3 and planned to turn right at the intersection.

Table 6. Signalized intersection filter – vehicle turn data.

Field	Interpretation
`Time`	Seconds since simulation start.
`Light Cycle`	Length of traffic control cycle in seconds.
`#left turns`	Number of vehicles that turned left during the 30-second interval.
`#ahead`	Number of vehicles that went straight during the 30-second interval.
`#right turns`	Number of vehicles that turned right during the 30-second interval.
`#total lost`	Number of vehicles that were off-plan during the 30-second interval.
`#lost with plan to turn left`	Number of vehicles that were off-plan and planned to turn left during the 30-second interval.
`#lost with plan to go straight`	Number of vehicles that were off-plan and planned to go straight during the 30-second interval.
`#lost with plan to turn right`	Number of vehicles that were off-plan and planned to turn right during the 30-second interval.

2. SCENARIOS

2.1 Multimode Network

2.1.1 Overview

The Multimode Network is constructed to test the simultaneous use of multiple travel modes and a set of feedback selectors. It is used to demonstrate the transfer of travelers from the *walk* network to either a *drive* or a *transit* network. Additionally, it allows the entire TRANSIMS framework to be exercised. Multimode is a layered network with a roadway, walkway, and a transit layer. The roadway and walk layers each form a circle with, for display purposes, the walk network set considerably inside the drive network. This leads to some non-realistic transfer times (walk times) between the walk and the other layers of the network, but does not interfere with the main purpose of this network.

The simplified population and activity generators may be used to generate activity sets on the Multimode Network. The Simplified Population Generator produces single-person households that are located on the *walk* layer of the network. The Simplified Activity Generator assigns three activities to each person in the household. These are a sequence of home-work-home activities, where the work location is determined using the land-use data assigned to the activity location.

2.1.2 Description

There are three network layers in the Multimode Network as shown in Fig. 4.

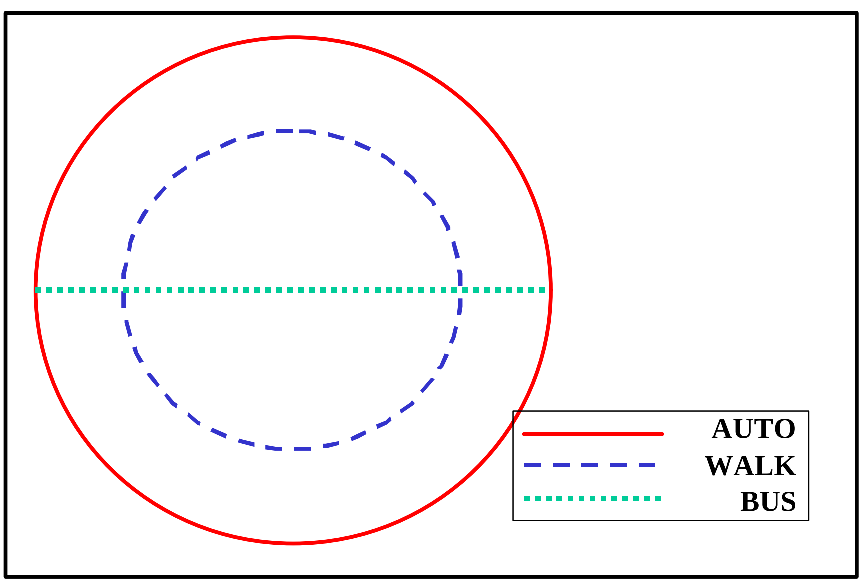


Fig. 4. The Multimode Network has three layers. The auto and walk networks are in a circle. A bus only layer runs across the circle. The layers are connected by process links.

There are 100 links on both the auto and the walk networks. The link numbers on the auto network are 1 through 100. The walk links are numbered 1001 through 1100. Each of these links has either a parking location (auto links) or an activity location (walk links). The parking location numbers are 1 through 100, corresponding to the auto link numbers. Similarily, the activity locations are numbered 1001 through 1100. There are three links on the bus network, and these are numbered 10001 through 10003.

The driving or roadway links are each 750 meters (100 cells) long. They have one lane in each direction with a maximum speed of 37.5 meters (five cells) per second. The walk links are each 375.9 meters long. The bus links are bidirectional, one-lane roads that allow for bus traffic only. The links 10001 and 10003 are 76 meters long, while link 10002 is 23,724.2 meters.

The bus links are shown in Fig. 5. These bidirectional links are restricted to buses only, and buses run on a five-minute schedule in both directions. Buses originate from the bus yards (Yard-3003 and Yard-3100) on Links 10001 and 10003. Their routes cover only the three Links 10001, 10002 and 10003. Each bus makes two stops to load and unload passengers at the bus stops, Stop-3002 and Stop-3001 or Stop-3101 and Stop-3102.

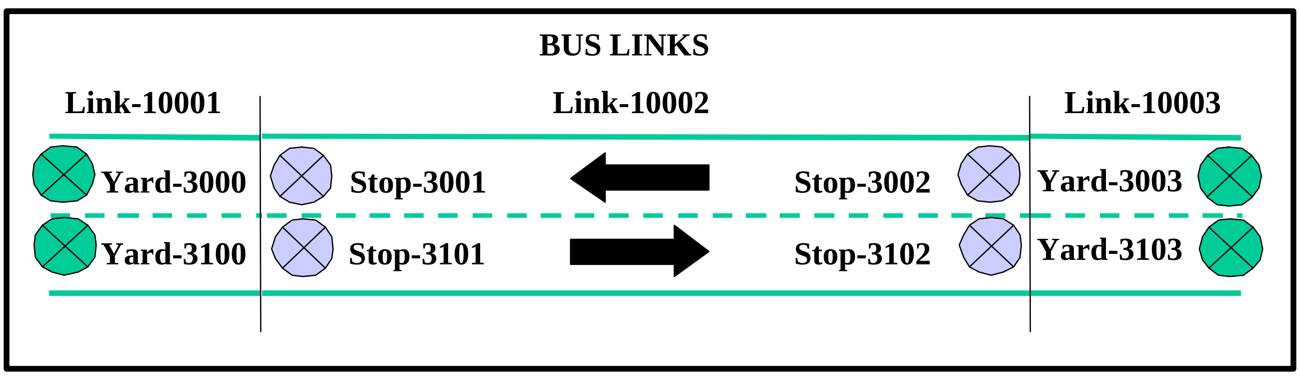


Fig. 5. The three bidirectional bus links for the Multinode Network. Buses start and end at the yards and stop to load and unload passengers at the two bus stops on link 10002.

Each activity location is connected to the corresponding parking location (e.g., Parking Location 2 is connected to Activity Location 1002) by two process links. One process link allows travelers to move from the activity location to the parking location, while the other allows movement in the opposite direction. The delay time associated with each process link is 10 seconds. A pair of process links is illustrated in Fig. 6.

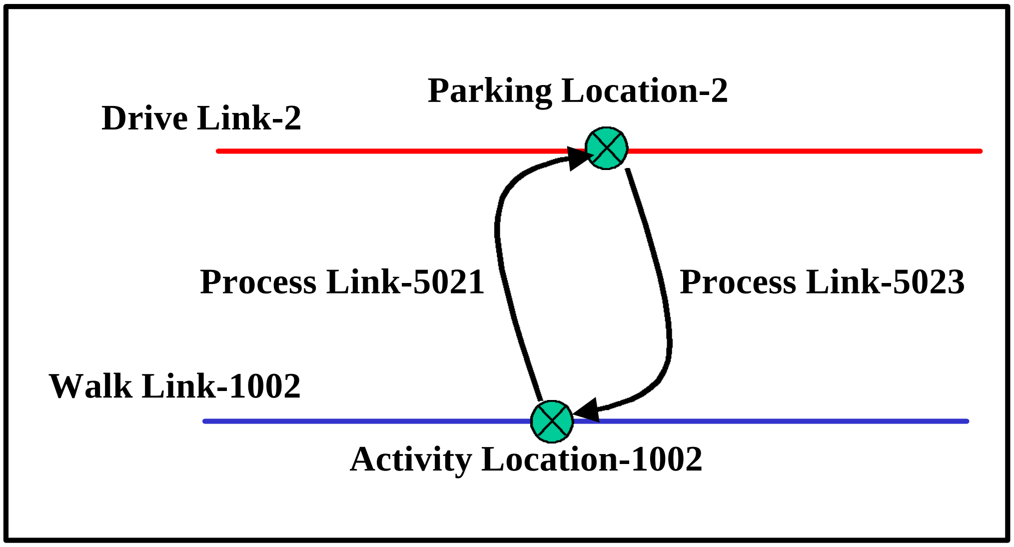


Fig. 6. The process links connecting Activity Location 1002 to Parking Location 2.

The process link numbers to connect parking location *p* to activity location, 1000+*p*, are computed as:

$5000+10*p+1$ for moving from activity to parking

and

$5000+10*p+3$ for moving from parking to activity

Transit stops are connected to the nearest activity locations on the walk network by a series of process links. Fig. 7 shows the process links that connect transit Stop-3001 and Stop-3002 to the Activity Locations 1050 and 1051. A similar set of process links connects Activity Locations 1001 and 1100 to transit Stop-3101 and Stop-3102.

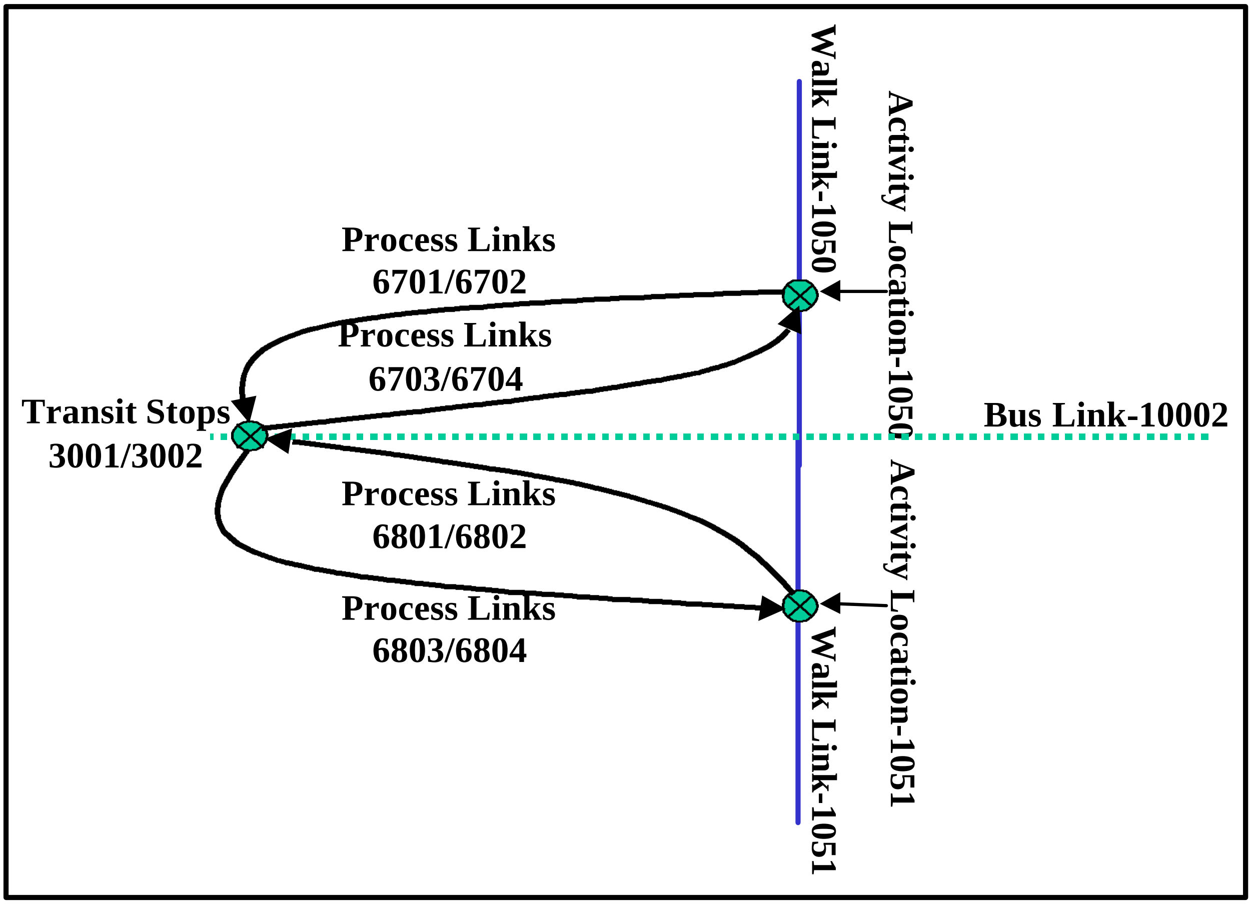


Fig. 7. Process links connect the transit stops with activity locations on the walk network.

All activity locations on this network are on the *walk* layer. Three land-use variables are associated with each activity location: *A*, access to transit; *H*, a weight for placing households at the location; and *W*, a weight for work locations. Each value of *A* gives the walking distance in meters to the nearest transit stop. The values of *W* and *H* alternate between the values 0 and 1. Odd activity locations have *H* set to 1 and *W* set to 0. Even activity locations have *W* set to 1 and *H* set to 0.

2.1.3 Usage

The Multimode Network is used to demonstrate feedback through the TRANSIMS components. The Simplified Population Generator, Simplified Activity Generator, Route Planner, and Traffic Microsimulator are used in the iterations. The feedback process is initiated and controlled by an iteration script. The sample script supplied with this distribution in *$TRANSIMS_HOME/data/calnet/scripts/calnet.csh* is shown below. Output to the output subdirectory generated by this script is ~350 Megabytes.

```
#!/bin/csh

#  Usage:  $TRANSIMS_HOME/data/calnet/scripts/calnet.csh
```

```
#  This script generates buses and travelers for calnet
#  using the simplified population and activity generators.
#  New activities are generated for travelers with unrealistic
#  plans or schedules as identified by the router; this
illustrates
#  router/activity generator feedback.  After the microsimulator
#  is run, additional activity generator feedback is provided
#  by the selector.

#  This script assumes the following:

#   TRANSIMS_HOME is set, and the directory structure below
#      in $TRANSIMS_HOME/data/calnet is as in the release.
#   csh resides in /bin
#   awk and pvm are in your path
#   the user has write permission in the output subdirectory

#  Step (0).  Initialization.

set CONFIG=$TRANSIMS_HOME/data/calnet/calnet_default.cfg
set OUTPUT=$TRANSIMS_HOME/data/calnet/output
set BIN=$TRANSIMS_HOME/bin
set LOGFILE=$OUTPUT/runlog

set ACTIVITIES=$OUTPUT/activities
set PLANS=$OUTPUT/plans

echo "quit" | pvm –

#  Step (1).  Generate bus routes, schedules, vehicles and plans.

$TRANSIMS_HOME/data/calnet/scripts/GenCalnetBuses.sh \
     $PLANS.bus $OUTPUT/vehicles.bus $OUTPUT/transit.schedule \
     $OUTPUT/transit.route 9000 32400

#  Step (2).  Generate initial activities and plans.

#  Step (2a).  Generate synthetic population with vehicles.

$BIN/Popgen $CONFIG >& $OUTPUT/Popgen.errs
$BIN/Poploc $CONFIG >& $OUTPUT/Poploc.errs
$BIN/Vehgen $CONFIG >& $OUTPUT/Vehgen.errs
awk 'NR>1' $OUTPUT/vehicles.bus >> $OUTPUT/vehicles

#  Step (2b).  Generate initial activities.

$BIN/Actgen $CONFIG >& $OUTPUT/Actgen.errs
mv $ACTIVITIES $ACTIVITIES.0
$BIN/IndexFilenames $ACTIVITIES.hh.idx w names
sed 's/activities/activities.0/' names >! newnames
$BIN/IndexFilenames $ACTIVITIES.hh.idx r newnames
$BIN/IndexFilenames $ACTIVITIES.trv.idx w names
```

```
sed 's/activities/activities.0/' names >! newnames
$BIN/IndexFilenames $ACTIVITIES.trv.idx r newnames

#  Step (2c).  Generate initial plans and problem file.

$BIN/Router $CONFIG >& $OUTPUT/Router.errs
mv $PLANS $PLANS.0
cat $PLANS.bus >> $PLANS.0
$BIN/IndexPlanFile $PLANS.0
mv $PLANS.0.trv.idx $PLANS.trv.idx
mv $PLANS.0.tim.idx $PLANS.tim.idx
mv $OUTPUT/router.problems $OUTPUT/router.problems.0

#  Step (3).  Feedback loops from router to activity generator

set iter=0
while ($iter < 2)

#  Step (3a).  Select travelers who found no path.

   awk '$4 < 0 {print $2 " L"; print $2 " M"}' \
      $OUTPUT/router.problems.$iter | sort –n | uniq \
      >! $OUTPUT/activity.feedback
   awk '$4 < 0 {print $1}' $OUTPUT/router.problems.$iter \
     | sort –n | uniq >! $OUTPUT/router.feedback

#  Step (3b).  Generate new work locations and mode preferences.

   @ iter++
   rm $OUTPUT/ActRegen.errs
   $BIN/ActRegen $CONFIG >& $OUTPUT/ActRegen.errs
   mv $ACTIVITIES.partial $ACTIVITIES.$iter
   $BIN/IndexActivityFile $ACTIVITIES.$iter
   mv $ACTIVITIES.hh.idx tindex
   $BIN/MergeIndices $ACTIVITIES.hh.idx tindex
$ACTIVITIES.$iter.hh.idx
   rm $ACTIVITIES.trv.idx

#  Step (3c).  Generate new plans.

   rm $OUTPUT/Rerouter.errs
   $BIN/Router $CONFIG >& $OUTPUT/Rerouter.errs
   mv $PLANS.trv.idx tindex
   mv $PLANS $PLANS.$iter
   $BIN/IndexPlanFile $PLANS.$iter
   $BIN/MergeIndices $PLANS.trv.idx tindex $PLANS.$iter.trv.idx
   rm $PLANS.tim.idx
   mv $OUTPUT/router.problems $OUTPUT/router.problems.$iter

end
```

```
#  Step (4).  Feedback loop from microsimulation to activity
generator.

set jter=0
while ($jter < 2)

#  Step (4a).  Run the microsimulator.

   $BIN/CA $CONFIG>!$OUTPUT/CAlogfile.$jer
   cat $OUTPUT/logfile >> $OUTPUT/CAlogfile.$jter

#  Step (4b).  Run the selector.

   rm $OUTPUT/Selector.errs
   $BIN/Selector $CONFIG >& $OUTPUT/Selector.errs

#  Step (4c).  Add travelers who failed to reach their
destination to
#  the activity and router feedback files for relocation and
remoding.

   awk –F, ‘NR>2 && $3==2 && $11 == 0 {print $1, $2}’ \
       $OUTPUT/itdb.00$jter.it >! temp
   awk –F, ‘NR>2 {print $1, $2}’ $OUTPUT/itdb.00$jter.it \
       | sort –n | uniq –c | awk ‘$1 == 1 {print $2, $3}’ >> temp

   awk ‘$1 < 9000 {print $1 “ L”; print $1 “ M”}’ temp \
       >> $OUTPUT/activity.feedback
   sort –n $OUTPUT/activity.feedback | uniq >! sorted
   sed ‘s:T:T-240:’ sorted >! $OUTPUT/activity.feedback

   awk ‘$1 < 9000 {print $2}’ temp >> $OUTPUT/router.feedback
   sort –n $OUTPUT/router.feedback | uniq >! sorted
   mv sorted $OUTPUT/router.feedback

#  Step (4d).  Generate new activities.

   @ iter++
   cp $OUTPUT/activity.feedback $OUTPUT/activity.feedback.$iter
   rm $OUTPUT/ActRegen.errs
   $BIN/ActRegen $CONFIG >& $OUTPUT/ActRegen.errs
   mv $ACTIVITIES.partial $ACTIVITIES.$iter
   $BIN/IndexActivityFile $ACTIVITIES.$iter
   mv $ACTIVITIES.hh.idx tindex
   $BIN/MergeIndices $ACTIVITIES.hh.idx tindex
$ACTIVITIES.$iter.hh.idx
   rm $ACTIVITIES.trv.idx

#  Step (4e).  Generate new plans.

   rm $OUTPUT/Rerouter.errs
   $BIN/Router $CONFIG >& $OUTPUT/Rerouter.errs
```

```
    mv $PLANS.trv.idx tindex
    mv $PLANS $PLANS.$iter
    $BIN/IndexPlanFile $PLANS.$iter
    $BIN/MergeIndices $PLANS.trv.idx tindex $PLANS.$iter.trv.idx
    rm $PLANS.tim.idx
    mv $OUTPUT/router.problems $OUTPUT/router.problems.$iter

    @ jter++

end

#  Step (5).  Make a snapshot file for the final iteration

$BIN/indexvehtobin $OUTPUT/snapshot.veh $OUTPUT/snapshot.veh.bin
```

Step (0) sets some local variables and checks that PVM is running. The path names must agree with the values for these keys provided in the configuration file, and PVM is assumed to be in the user's path. In the configuration file supplied with this release, *calnet_default.cfg*, all paths are referenced to a *TRANSIMS_ROOT* directory, which is the same directory as *TRANSIMS_HOME*.

The initial iteration through the components (iteration 0) will generate the following files in the output directory.

- *Transit (bus) routes and schedules, plus vehicles and plans for the bus drivers (Step (1)).* The path names to the bus route and schedule files are those specified in the configuration file. Buses run in both directions along the central bus line, departing every five minutes, from 2:30 (earlier than any traveler is scheduled to leave for work) until 9:00 (later than the last traveler plans to return home).

- *A synthetic population composed of single-person households located on the Multimode Network together with a vehicle file for the population (Step (2a)).* The vehicle file containing the bus vehicles is appended to the population vehicle file. The Simplified Population Generator and Simplified Population Location discussed in Section 2.1.4 are used.

- *Home-work-home activities for the population (Step (2b)).* See Section 1.7.5 for a description of the scenario used by the Simplified Activity Generator. In particular, travelers are scheduled to arrive at work between 3:00 and 3:30 and to remain there for approximately three hours before returning home. The activity file created by the Simplified Activity Generator is moved to *activities.0* in the output directory. The indexes for the activity file (which hereafter exists only as a notional file) are modified using IndexFilenames.

- *Route plans for the population activities (Step (2c)).* Again, the resulting plan file, *plans*, is moved to *plans.0*. The plans for the bus drivers are appended to *plans.0*.

In this initial pass, the Route Planner identifies a number of problems. In particular, more than half of the travelers assigned to ride the bus are unable to make plans for returning home before bus service ends. This occurs because their good "access" to transit (which is the basis for assigning them to transit when `ACT_MODE_CHOICE_OPTION = 4`) is the result of living and working near the same bus stop on the simple Multimode Network, a condition for which the Simplified Activity Generator does not check. These problematic travelers are identified in Step (3a), and used to generate activity feedback files (based on traveler ID) and Route Planner feedback files (based on household ID). The Simplified Activity Regenerator is used to select new work locations and new mode preferences for the selected travelers (Step (3b)). The output of this component, which is a partial activities file containing only activities for the reprogrammed travelers, is saved as *activities.1*, and a merged index file is created. Similarly, the Route Planner replans only the households listed in the Route Planner feedback file (Step (3c)). These new plans are saved as *plans.1*, and another merged index file is created. Step (3) is repeated a second time to weed out any remaining problems.

Thus at the beginning of Step (4), the main feedback loop, the activity file indexes point to *activities.2*, *activities.1* and *activities.0*, while the plan file indexes point to *plans.2*, *plans.1* and *plans.0*. Each iteration of the main feedback loop begins with a run of the Traffic Microsimulator (Step (4a)). Simulation covers the time period 2:45 to 4:45, which is sufficient to get most of the travelers to work but does not include the return trip. The output of the Traffic Microsimulator is contained in event, summary, and snapshot files in the output directory.

Step (4b) uses the Selector to create the iteration database as well as activity and router feedback files. The iteration database contains summary results for trips 1 and 2 for most travelers, including expected and actual durations, total and geometric distances, etc. These statistics are used to compute several cost functions, as described in the section on the Selector, which in turn are the basis for its selection of travelers to relocate, remode, and retime. To the travelers identified by the Selector, Step (4c) adds travelers who failed to complete trip 2. This is indicated by either no entry for trip 2, or an entry in which the total time is zero. These travelers will again receive both new work locations and new mode preferences in Step (4d). For travelers selected by retiming, the time to leave work estimated by the Simplified Activity Regenerator is moved back five minutes. Finally, new plans are generated for the reprogrammed travelers in Step (4e).

The main iteration loop is repeated two more times in this script. Each iteration overwrites the event, summary, and snapshot files in the output directory. Thus Step (5) sees only the most recent summary file, for which it creates a binary snapshot file that is required by the Output Visualizer, using the *snapshot.veh* file created by the last iteration. The tutorial in Volume Three (*Modules*), Chapter Eight (*Output Visualize)* has instructions for running the Output Visualizer to view output of the TRANSIMS modules.

2.1.4 Simplified Population Generator

Two programs are used to produce a simplified synthetic population located on the multimode network. A TRANSIMS configuration file is used to specify parameters to the programs. Entries in the configuration file are of the form `<KEY> <value>`

2.1.4.1 Generating a Simplified Baseline Population

Popgen is the program that generates a baseline population of single person households with one vehicle per household. This population is not located on the transportation network and uses a value of –1 for the household location. *Popgen* assigns household and person ids and generates income as the only person demographic in the population.

Usage

```
% $TRANSIMS_HOME/bin/Popgen <configuration file>
```

Popgen uses the following configuration file keys (Table 7) from the TRANSIMS configuration file. Some keys have default values that may be used if the key is not specified in the configuration file.

Table 7. Popgen configuration file keys.

Configuration File Key	Description
POP_NUMBER_HH	The number of households to be generated.
POP_BASELINE_FILE	The name of output file.
POP_STARTING_HH_ID	The number from which the generated households will be sequentially numbered. Default = 1
POP_STARTING_PERSON_ID	The number from which the generated persons will be sequentially numbered. Default = 101

2.1.4.2 Simplified Population Locator

Two programs are available to locate a population on a transportation network, *Poploc* and *BlockGroupLocator*. *Poploc* should be used when the baseline population was generated independent of census tract and block group information. The TRANSIMS Simplified Population Generator (*Popgen*) produces a population of this type. *Poploc* is the program that locates a population on a transportation network based upon choice of a home location from residential land use values specified in the network activity location file (user data). All activity locations in the network that have residential values greater than zero are candidates for the selection. The baseline population should contain valid household and person ids since *Poploc* does not generate them.

Usage

```
% $TRANSIMS_HOME/bin/Poploc <configuration file>
```

Poploc uses the following configuration file keys (Table 8) from the TRANSIMS configuration file. Some keys have default values which may be used if the key is not specified in the configuration file.

Table 8. Poploc configuration file keys.

Configuration File Key	Description
POP_BASELINE_FILE	The name of file containing the baseline population.
POP_LOCATED_FILE	The name of file where the located population will e written.

Configuration File Key	Description
ACT_HOME_HEADER	The user data column header in the network activity location file used to specify household home locations. Default = HOME
NET_DIRECTORY	The directory where the network files reside.
NET_NODE_TABLE	The network node table name.
NET_LINK_TABLE	The network link table name.
NET_ACTIVITY_LOCATION_TABLE	The network activity location table name

2.1.5 Simplified Activity Generator

2.1.5.1 Overview

We use a simplified generator that has all of the characteristics of the full-version Activity Generator to produce activities for the small Multimode Network that has activity locations around a circle. This Simplified Activity Generator makes a full activity file.

Activities are placed at the activity locations on the network, where all of the activity locations are located on the *walk* layer of the small Multimode Network. Each individual in the simplified population for this network is assigned three activities:

- the individual starts with an activity at home,
- followed by one at work, then
- a return activity at home.

Three options are given for the location of the work activity. A mode choice model is also given, with multiple options for determining the mode of travel. Table 9 lists the simplified activity generator configuration file keys.

Table 9. Simplified activity configuration file keys.

Configuration File Key	Description
ACT_MODE_ALPHA	The alpha parameter used to generate mode choice in the simplified activity generator.
ACT_MODE_BETA	The beta parameter used to generate mode choice in the simplified activity generator.
ACT_MODE_CHOICE_OPTION	The option used to select the mode choice algorithm in the simplified activity generator.
ACT_TIME_ALPHA	The alpha parameter used to generate activity times in the simplified activity generator.
ACT_TIME_BETA	The beta parameter used to generate activity times in the simplified activity generator.
ACT_WORK_LOC_ALPHA	The alpha parameter used to generate work locations in the simplified activity generator.
ACT_WORK_LOC_BETA	The beta parameter used to generate work locations in the simplified activity generator.

Configuration File Key	Description
ACT_WORK_LOC_GAMMA	The gamma value used to generate work locations in the simplified activity generator.
ACT_WORK_LOCATION_OPTION	The option used to select the work location algorithm in the simplified activity generator.

2.1.5.2 Determining the Work Location

This option is based on a general gravity method and has input parameters *a*, *b*, and *g*, all of which are greater than 0. Let $d_{I,j}$ be the distance from activity location *I* to activity location *j*. Then, for a fixed home location, *I*, the following weight $h_{I,j}$ is computed as

$$h_{ij} - I(W_j)\{b(Exp(-ad_{ij}/W_j) + g\}$$

where W_j is the work entry from the activity location file and both $I(W_j)$ and $h_{I,j}$ equal 0 if W_j is 0. Then, the probability of a location *j* being the work location is

$$p_j = h_{ij} / \sum_k h_{ik}$$

The work location is determined by drawing a random number between zero and one, *z*, and selecting activity location I as the work location if

$$\sum_{k=1}^{i-1} p_i < z \leq \sum_{k=1}^{i} p_i$$

2.1.5.3 Determining Activity Times

All times in the activity file are in hours and fractions of hours. We assume that the first home activity starts at time 0 and that all work activities start between 0.5 and 3.5 hours (1,800 and 12,600 seconds) after time 0. In addition, the work duration is between 2.778 and 5 hours (10,000 and 18,000 seconds). The time spans in the activity file are all ±5 minutes or 0.083 hours. The parameters *a* and *b* in the file are set to –1.

The time to start work, T_W, depends on two input parameters, *a* and *b*. Other activity times, L_H, the time to leave home, *D*, the work duration, T_L, the time to leave work, and T_H, the expected arrival at home, are all based on the initial time to start work, T_W. These quantities are computed as follows:

- The time to start work: $T_W = (10800\ \beta(\alpha,\beta) + 1800)/3600$, where $\beta(\alpha,\beta)$, represents a random draw from a beta distribution, with parameters *a* and *b* as described in the section on simplified population generation.
- The time to leave home: $L_H = T_W - d_{I,j} / (4.8*7.5*3600)$. Note that $d_{I,j}/(4.8*7.5*3600)$ is the average time in hours that it takes to travel on a non-congested road between

parking locations *I* and *j*. "Cells" in the microsimulation are 7.5 meters long and the average speed is 4.8 "cells" per second.

- The work duration: $D = (8000 * z + 10000)/3600$, where *z* is a random number between zero and one.
- The time to leave work: $T_L = T_W + D$.
- The time to arrive home: $T_H = T_L + d_{I,j} / (4.8*7.5*3600)$.

Table 10 provides these quantities.

Table 10. Activity time table.

Activity	Start-a	Start-b	End-a	End-b	Duration-a	Duration-b
Home	0	0	L_H-.083	L_H+.083	L_H-.083	L_H+.083
Work	T_W-.083	T_W+.083	T_W-.083+D	T_W+.083+D	D-.167	D+.167
Home	T_H-.083	T_H+.083	24.	24.	24- T_H-.083	24- T_H+.083

2.1.5.4 Determining Travel Modes

The following three mode sequences are allowed for the activities on the small Multimode Network:

w: Walk only

wcw: Walk-Car-Walk

wtw: Any combination of Walk and Bus

The Simplified Activity Generator has three options for determining mode of transportation. One of these options, Option Four, is a parameterized mode choice model in which accessibility to transit is considered. The other options force driving or randomly select the modes.

Option One In this option, all modes are Walk-Car-Walk, mode sequence 2 (wcw).

Option Two One of the mode sequences (w, wcw, or wtw) is randomly selected.

Option Three All modes are transit modes. A transit mode string (wtw, wbw) must be specified in the mode map file.

Option Four In this option, we use a simplified mode choice model to select either the mode sequence (wcw) or any combination of (wtw). The selection is based on the traveler's accessibility to transit. The activity location file for the Multimode Network contains a column denoted "`ACCESS`" for accessibility. The entries in this column, A_I, are the walking distances in meters to the nearest transit stop.

The mode choice model depends on two user-supplied parameters: $\alpha \geq 0$ and $0 \leq \beta \leq 2$. If a traveler is traveling between activity location *I* and activity location *j*, the probability of his taking transit (mode sequence 3) is given by the following equation:

$$p = \hat{a}/(1 + \exp\{\acute{a}\ (A_i + A_j)\})$$

The choice between wcw and wtw is then determined by drawing a random number, *z*, between zero and one. If *z* is less than *p*, mode sequence 3 is chosen. Otherwise, mode sequence 2 (an automobile trip) is selected.

2.1.5.5 Usage

There are two computer programs that can generate or modify simplified activities. The first is known as *Actgen*; it generates simplified activity sets for all the individuals in the synthetic population. The second is known as *ActRegen*, and it modifies certain attributes of activities for travelers specified in a TRANSIMS feedback file. Table 11 provides common configuration file keys for both programs.

Table 11. Actgen and ActRegen common configuration file keys.

Configuration File Key	Description
ACT_ACCESS_HEADER	The column header in the activity location file for transit accessibility. Default = ACCESS
ACT_HOME_HEADER	The column header in the activity location file for single family home locations. Default = HOME
ACT_MODE_ALPHA	The alpha value used to generate activity mode choice (gravity method). Default = 1.0
ACT_MODE_BETA	The beta value used to generate activity mode choice (gravity method). Default = 1.0
ACT_MODE_CHOICE_OPTION	The option used to select the mode choice algorithm (1 – 4). Default = 2 (random mode choice).
ACT_RANDOM_SEED	Random number seed. Default = 985456379
ACT_TIME_ALPHA	The alpha value used to generate activity times. Default = 1.0
ACT_TIME_BETA	The beta value used to generate activity times. Default = 1.0
ACT_WORK_HEADER	The column header in the activity location file for work locations. Default = WORK
ACT_WORK_LOC_ALPHA	The alpha value used to generate work locations. Default = 1.0
ACT_WORK_LOC_BETA	The beta value used to generate work locations. Default = 1.0
ACT_WORK_LOC_GAMMA	The gamma value used to generate work locations. Default = 1.0
ACT_WORK_LOCATION_OPTION	The option used to select the work location algorithm (1 – 3). Options 2 and 3 are not implemented in this release. Default = 1

Configuration File Key	Description
MODE_MAP_FILE*	The mode file that defines a correspondence between TRANSIMS mode strings; i.e., wcw and an integer value that is used in a TRANSIMS activity file.
NET_ACTIVITY_LOCATION_TABLE*	The network activity location table name.
NET_DIRECTORY*	The directory where the network files reside.
NET_LINK_TABLE*	The network link table name.
NET_NODE_TABLE*	The network node table name.
POP_LOCATED_FILE*	The name of the file containing the located population.
VEHICLE_FILE*	The name of the TRANSIMS vehicle file for the population.

* Required configuration file keys. All others are optional and will use default values.

Actgen

This program generates simplified activities for each person in a located population file. Three activities are generated for each person (home, work, home). Times (start, end, and duration), work location, and mode choice are determined for each set of three activities. The program uses several probability distribution parameters. Values for these parameters, as well as other program options, are set using TRANSIMS configuration file keys.

Usage

```
% $TRANSIMS_HOME/bin/Actgen <configuration file>
```

Table 12 shows the configuration file key from the TRANSIMS configuration file (in addition to the keys found in Table 11).

Table 12. Actgen configuration file key.

Configuration File Key	Description
ACT_FULL_OUTPUT*	The name of the file where the generated activities will be written.

* Required configuration file keys. All others are optional and will use default values.

ActRegen

This program regenerates leave-home time, work locations, or mode choices for a traveler’s activities. Traveler IDs to be regenerated and the type of information to be regenerated are read from a TRANSIMS feedback file with the following format:

```
<traveler id>  <feedback command>
```

Table 13 lists the valid feedback commands for *ActRegen*.

Table 13. ActRegen feedback commands.

Feedback Command	Description
L	Generates a new work location.
T<value in seconds>	Generates a new time to leave home for work. If <value> is given, add the given value to the new time generated. Value may less than zero.
M	Generates a new mode choice for the work and return to home activities.

Usage:

```
% $TRANSIMS_HOME/bin/ActRegen <configuration file>
```

In addition to the configuration file keys listed in Table 13, *ActRegen* uses the configuration file keys from the TRANSIMS configuration file. These are listed in Table 14.

Table 14. ActRegen configuration file keys.

Configuration File Key	Description
ACT_FEEDBACK_FILE*	The name of the file containing feedback commands.
ACT_PARTIAL_OUTPUT*	The name of the file where the regenerated activities will be written.
ACTIVITY_FILE *	The name of the file that contains the previous activity set.

* Required configuration file keys. All others are optional and will use default values.

2.2 Toynet Network

2.2.1 Overview

The Toynet Network is constructed to test feedback between the Route Planner and the Traffic Microsimulator. It is a single layer, gridded network with several zones. The Trip Table Activity Generator is used to construct trips between zones. The iteration script provided with this release illustrates the use of a default configuration file and the modification of a handful of keys between iteration according to a schedule specified by the user.

2.2.2 Description

There are several types of links in the Toynet Network as shown in ***Fig. 8***. The twelve freeways, three on each side, extend out five kilometers and are connected to the central section by one kilometer of primary artery. All of the remaining links are 500 meters long. Freeways and primary and secondary arteries have two lanes in each direction. Speed limits are 33.75 m/s on the freeway, 30 m/s on the primary arteries, and 22 m/s on the secondary arteries. Collector and local streets, with one lane in each direction, have speed limits of 15 m/s.

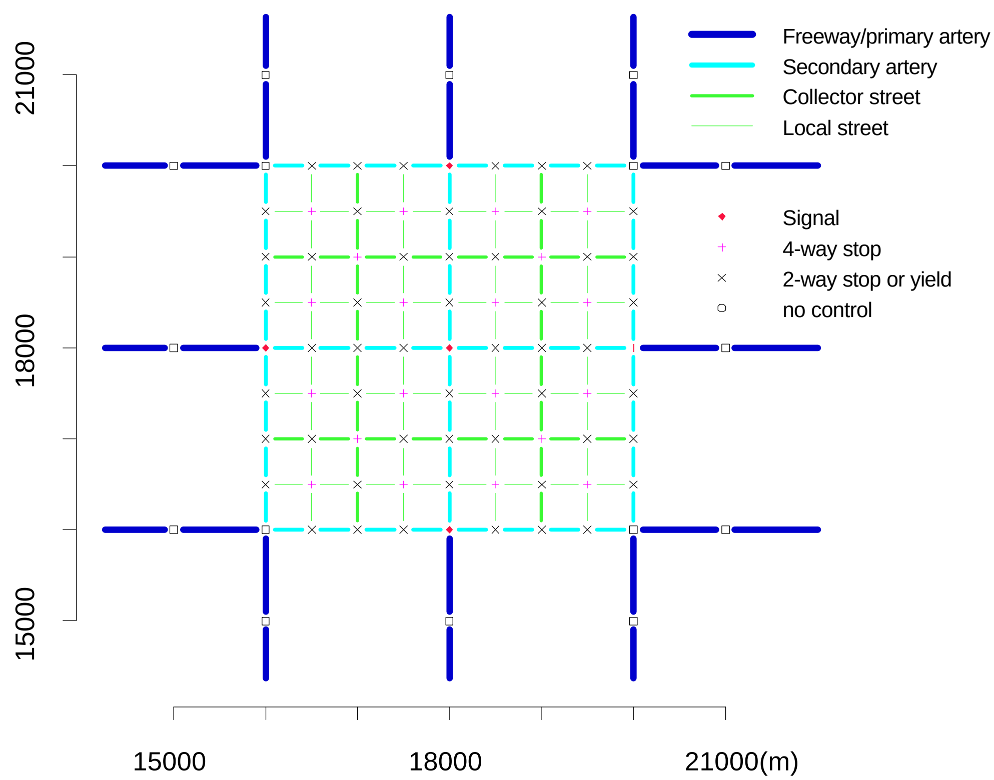


Fig. 8. Toynet Network links and nodes.

- The arteries have left-turn pocket lanes where they intersect other arteries. Timed signals are provided at the five central intersections as shown in Fig. 8. Toynet Network links and nodes.

- . Other intersections have stops, yields, or no controls.

- Twenty-four BNDRY-style parking lots with associated activity locations are located 50 meters from the end of each freeway, one on each side of the freeway. There are no parking or activity locations on the primary arteries. The remaining links have activity locations and parking lots midway between their end nodes on each side of the link.

- The interior of each of the four main subblocks constitutes a zone (4 zones), defined in the Toynet ACTIVITY_LOCATION_TABLE, as shown in Fig. 9. Toynet Network zones. Arrows show the links from which traffic enters the network and by which it leaves the network in the toynet.csh scenario.

- Each segment of a secondary artery bounding such a block is a separate zone (12 zones). Finally, each activity location at the ends of the freeway/primary artery links is in its own zone (12 end zones.) Thus numerous types of trips can be planned on this network: through traffic between end zones, local traffic among the central zones, trips between "home" locations in the interior of the subblocks and "business" locations on the secondary arteries or "work" locations at the end nodes, and combinations of these. The scenario for the *toynet.csh* script described in the following section has only through travelers (red arrows in Fig. 9.)

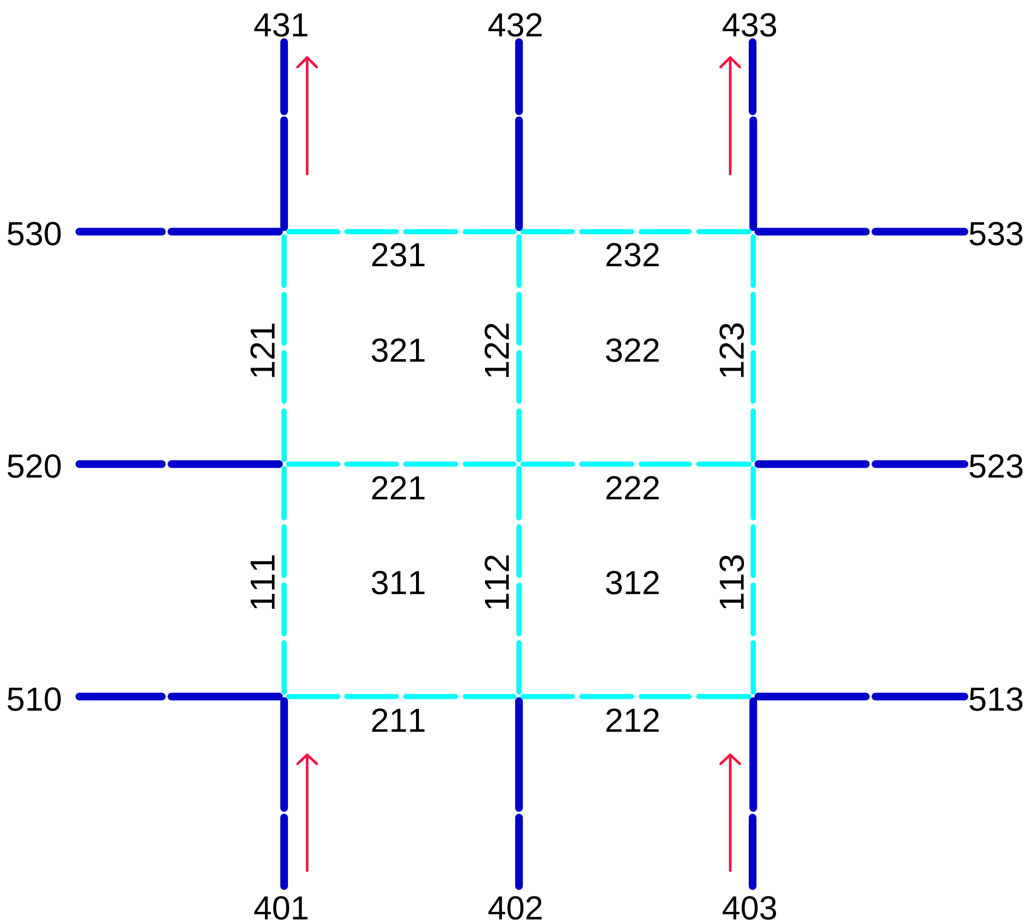


Fig. 9. Toynet Network zones. Arrows show the links from which traffic enters the network and by which it leaves the network in the toynet.csh scenario.

2.2.3 Usage

The Toynet Network is used to demonstrate feedback between the Route Planner and the Traffic Microsimulator. The Trip Table Activity Generator is used to generate the activity patterns to be simulated. The Route Planner, Traffic Microsimulator, and Selector are used in the iterations. A new four-line configuration file is generated for each iteration. Most of the configuration file keys are set in the default configuration file *toynet_default.cfg* to which this configuration file refers.

The feedback process is initiated and controlled by an iteration script. The sample script supplied with this distribution in *$TRANSIMS_HOME/data/toynet/scripts/toynet.csh* is shown below. Output to the output subdirectory generated by this script with niter=3 is ~240 Megabytes.

```
#!/bin/csh
#  Iteration on the toynet network.

#  The environmental variable TRANSIMS_HOME must be set.
```

```
#  This script assumes that the TRANSIMS_HOME directory is
#  structured like the TRANSIMS_HOME directory of this
#  release.  In particular, network, data, scripts and output
#  are subdirectories of $TRANSIMS_HOME/data/toynet.

#  Usage:
#      $TRANSIMS_HOME/data/toynet/scripts/toynet.csh niter
#  where
#      niter is the number of the last iteration

#  It is assumed that the output subdirectory is initially empty
#  and that the user has write permission to that subdirectory.

set niter = $1

#  (1) Check PVM
echo “quit” | pvm –

#  (2) Schedules for variable parameters
#  Router delay noise
set rdn=(0.4 0.2 0.1)
set nn=0
#  Selector rerouting fraction
set srf=(0.25 0.15 0.1 0.05)
set nf=0
#  Selector frustration threshold
set sft=(0.5 0.4 0.3)
set nt=0

#  (3) Pathnames
set CONFIG=$TRANSIMS_HOME/data/toynet/toynet.cfg

set OUTPUT=$TRANSIMS_HOME/data/toynet/output
set BIN=$TRANSIMS_HOME/bin

set VEHICLES=$OUTPUT/vehicles
set ACTIVITIES=$OUTPUT/activities
set PLANS=$OUTPUT/plans
set LOGFILE=$OUTPUT/runlog

#  (4) Population and activities
echo “Generate population and activities from trip tables” >! $LOGFILE
$BIN/ActTripGen $TRANSIMS_HOME/data/toynet/toynet_default.cfg >> $LOGFILE

#  (5) Router/CA iteration loop

set iter=0
while ($iter <= $niter)
   echo “” >> $LOGFILE
   echo “Starting iteration $iter” >> $LOGFILE
   if ($iter>0) then
      mv $PLANS.trv.idx $PLANS.last.trv.idx
   endif

#  (6) Create configuration file for this iteration
   if ($nn<$#rdn) then
       @ nn++
   endif
   if ($nf<$#srf) then
       @ nf++
   endif
   if ($nt<$#sft) then
       @ nt++
```

```
   endif
   echo “Router delay noise = $rdn[$nn]” >> $LOGFILE
   echo “Selector reroute fraction = $srf[$nf]” >> $LOGFILE
   echo “Selection frustration threshold = $thr[$nt]” >> $LOGFILE
   sed ‘s/RDN/’$rdn[$nn]’/;s/SRF/’$srf[$nf]’/;s/SFT/’$sft[$nt]’/’ \
       $TRANSIMS_HOME/data/toynet/data/config_iterate >! $CONFIG

#  (7) Plans
   echo “Generate new plans” >> $LOGFILE
   $BIN/Router $CONFIG >> $LOGFILE
   mv $PLANS $PLANS.$iter
   $BIN/IndexPlanFile $PLANS.$iter
   if ($iter>0) then
      MergeIndices $PLANS.trv.idx \
         $PLANS.last.trv.idx $PLANS.$iter.trv.idx
      rm $PLANS.tim.idx
   else
      mv $PLANS.$iter.trv.idx $PLANS.trv.idx
      mv $PLANS.$iter.tim.idx $PLANS.tim.idx
   endif

#  (8) Microsimulation
   echo “Run the microsimulator”  >> $LOGFILE
   echo “(Log info in $OUTPUT/CAlogfile.$iter)” >> $LOGFILE
   $BIN/ARCH.PVM.SUN4SOL2/CA $CONFIG >! $OUTPUT/CAlogfile.$iter
   cat $OUTPUT/logfile >> $OUTPUT/Calogfile.$iter

#  (9) Selector
   echo “Run the selector” >> $LOGFILE
   $BIN/Selector $CONFIG >> $LOGFILE

   @ iter++

end

#  (10) Create a binary snapshot file for the Visualizer

echo “Convert vehicle snapshot file for visualizer” >> $LOGFILE
$BIN/indexvehtobin $OUTPUT/snapshot.veh $OUTPUT/snapshot.veh.bin
```

The only command line argument for the *toynet.csh* script is `niter`, the number of the last iteration, a positive integer. The number of iterations will thus be `niter+1`, since the first iteration is iteration 0.

- Step (1) checks that pvm is running (pvm is assumed to be in the user’s path.)
- Step (2) defines lists of values for the variable keys to be changed at each iteration: `ROUTER_DELAY_NOISE`, `SEL_REROUTE_FRAC`, and `SEL_FRUSTRATION_THRESH`.
- Step (3) sets up some abbreviated path names that must agree with the values for these keys provided in the default configuration file. In the default configuration file supplied with this release, *toynet_default.cfg*, all paths are referenced to a *TRANSIMS_ROOT* directory which is the same as *TRANSIMS_HOME*.

The activity file is built in Step (4) by a call to the Trip Table Activity Generator using the default configuration file. (At this point, the file *toynet.cfg* to which $CONFIG points is not available.) The demand scenario is specified in the `ACT_TRIPTABLE_FILE` and

ACT_TRIPTIME_FILE. It calls for 1000 trips from zone 401 to zone 433 and 1000 from zone 403 to zone 431, with start times uniformly distributed from .05 to .5 hours past midnight.

Steps (5) through (9) compose the iteration loop, which is executed niter+1 times:

- Step (5) saves the old plan file index, which will be merged with the new one created at step (7).
- Step (6) creates *toynet.cfg* for this iteration, after updating the pointers into the lists of values for the variable keys.
- Step (7) routes the travelers through the network. The first iteration (iteration 0) constructs a plan file including all travelers. Subsequent iterations update plans only for travelers who have been selected by the preceding run of the Selector; their household ID numbers have been placed in the ROUTER_HOUSEHOLD_FILE by the Selector. The new plan files are saved as *$OUTPUT/plans.$iter*. Except on iteration 0, the index of the complete, notional plan file *$OUTPUT/plans*, sorted by traveler ID, is constructed using MergeIndices. The time-sorted index from the previous iteration must also be removed so that it will be reconstructed (using the up-to-date traveler-sorted index) by the Traffic Microsimulator. See Volume Five (*Software, Interface Functions and Data Structures*) Section 1.10 for a description of indexing and the indexing utilities including MergeIndices.
- Step (8) runs the Traffic Microsimulator with the new plans. Output event, summary, and snapshot files overwrite those created by preceding iterations.
- Step (9) updates the iteration database and selects travelers for rerouting in the following iteration. Refer to Volume Three (*Modules*), Chapter 6 (*Selectors/Iteration Databases*) for a description of the feedback process, iteration database, and the use of selectors demonstrated in this script.

The final Step (10) creates the binary snapshot file that is required by the Output Visualizer, using the *snapshot.veh* file created by the third iteration. The tutorial in Volume Three (*Modules*), Chapter Eight (*Output Visualizer*) has instructions on running the Output Visualizer to view output of the TRANSIMS modules.

2.3 Bignet Network

The Bignet (sample) Network and associated scenario have been designed to demonstrate TRANSIMS' principal components:

- Population Synthesizer,
- Activity Generator,
- Route Planner,
- Traffic Microsimulator (particularly cellular automata), and
- Emissions Estimator.

Although other programs will come into play during the course of the run, the modules noted above will be the principal players. Bignet also has been designed to demonstrate the computational resources required for a complete run of TRANSIMS using a small community (25,000 households).

Running Bignet is a computer-intensive effort. One-processor machines can run Bignet but obtaining results can take several days. Multi-computer systems with a slow network (less than 100 Mbps) or systems with slow read/write to storage media may take even longer to yield results. Memory requirements are also extensive, with maximum RAM usage on a one-CPU machine at approximately 1.7 gigabytes and even more for a multiprocessor machine. The final output will be approximately 1 gigabyte on your hard drive.

After installing TRANSIMS, you can use this stand-alone tutorial to get started. A working knowledge of Unix-based systems is beneficial but not mandatory.

2.3.1 Network Description

The Bignet Network consists of 3,853 nodes, 7,441 links with 1,7458 kilometers of roadway, and one parking location and two activity locations on almost every link. The Network is approximately one-tenth the size of Portland, Oregon, with 25,000 households containing 57,000 people, approximately 3,000 additional vehicles (itinerant travelers—those vehicles merely passing through the city), and another 3,000 freight vehicles.

Bignet exercises much of the code within the TRANSIMS modules; it has the normal population, trip-table activities, bus and rail transit, bicycles, bridges, a freeway, and restricted access roads. This version of Bignet does not incorporate feedback.

Bignet consists of an 8x8 grid of block groups, in which each block group consists of an 8x8 grid of local streets with secondary arterials on the edges. The following three sections

1) describe how land is used on the Network,
2) provide a detailed description of the Network, and
3) described the Network's connectivity.

Appendix A provides a detailed explanation of the naming convention used for Bignet's links and nodes.

2.3.1.1 Land Use in Bignet

The most basic way to view the Bignet Network is to look at its land use. Fig. 10 shows the Network's land use, zone divisions, and block groups. Fig. 11 in the next section shows how the streets are laid out within each block.

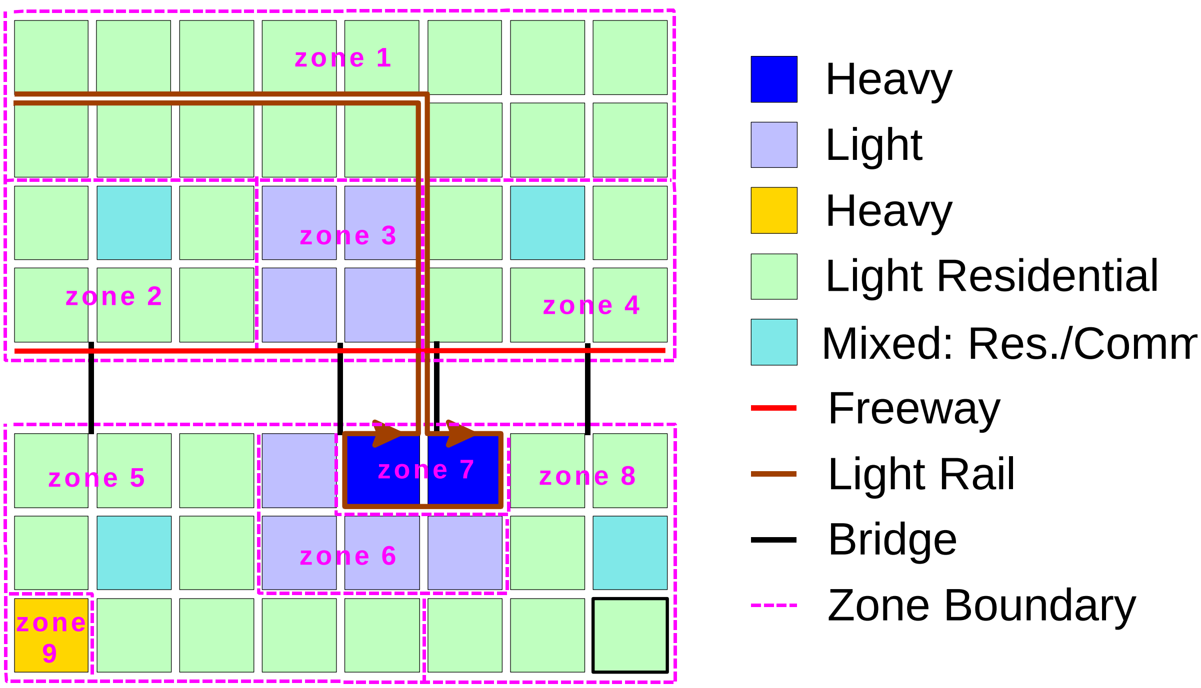


Fig. 10. In this schematic, the unlabeled white space across the middle of the network represents a river (the four black lines represent bridges). On the north side of town, just south of the local streets, is an east-to-west freeway. A single rail line runs from the northwest corner of the Network to the downtown area, around which it makes a loop and returns to the northwest station. Pink, dashed lines indicate zone boundaries, which are labeled with pink text.

Of the 15 zones in Bignet, nine are shown in Fig. 1. The remaining six consist of boundary zones. All activities of the basic grid shown in this figure are as follows:

- west is zone 10,
- north is zone 11,

- east is zone 12, and
- south is zone 13.

The freeways also have two zones: zone 14 west of the network and zone 15 east of the network.

Colors denote how land is used. For example, yellow shows that zone 9 is a heavy industrial area. This area has no homes but is the workplace for a significant fraction of the population. Because of its heavy industrial use, this zone is also a major freight location. Freight trips constitute the travel of approximately 5% of the population; these trips are made between both ends of the freeway and this industrial zone. There are also freight trips that never leave the freeway.

Colored dark blue, the "downtown" zone—like the heavy industrial zone—has no home locations but does serve as the workplace for much of the population. This area serves as the shopping and recreational destination for this community. Surrounding this downtown area and again on the north side of the river are two light commercial zones (light blue). These zones have the same features as the downtown area, except that the activities performed in these zones are far fewer than those in the downtown zone.

Most of the grid is colored light green, which represents residential zones. These block groups have nothing but home locations in varying densities. All travelers (except transit drivers, itinerant travelers, and freight) begin the simulation at time zero at a home location.

There are block groups in four of the primarily residential zones with both home locations and commercial activity locations. These zones (colored cyan) represent neighborhood stores, parks, strip malls, etc.

2.3.1.2 Detailed Description of the Network

Bignet consists of an 8x8 grid of block groups. Each block group has secondary arterials on its borders and local streets in between. The centermost local street in each direction is classified as a collector street.

On the left half of Fig. 2 is a diagram that shows only the links with a speed limit greater than two cells per second (15 m/s). The following color codes and their corresponding meanings are as follows:

Green = secondary arterials

Red = primary arterials

Black = freeway-endlinks

The freeway along the river and the rail-only links are not shown, although they both have a maximum speed of five cells per second (37.5 m/s).

The right half of Fig. 11 provides the detail of one block group with the following color codes and their corresponding meanings:

Green	=	secondary arterials
Dark blue	=	local streets
Light blue	=	collector streets

This diagram also shows all of the traffic signals in that local block group. Note that yield signs exist only on the outer edges of the network.

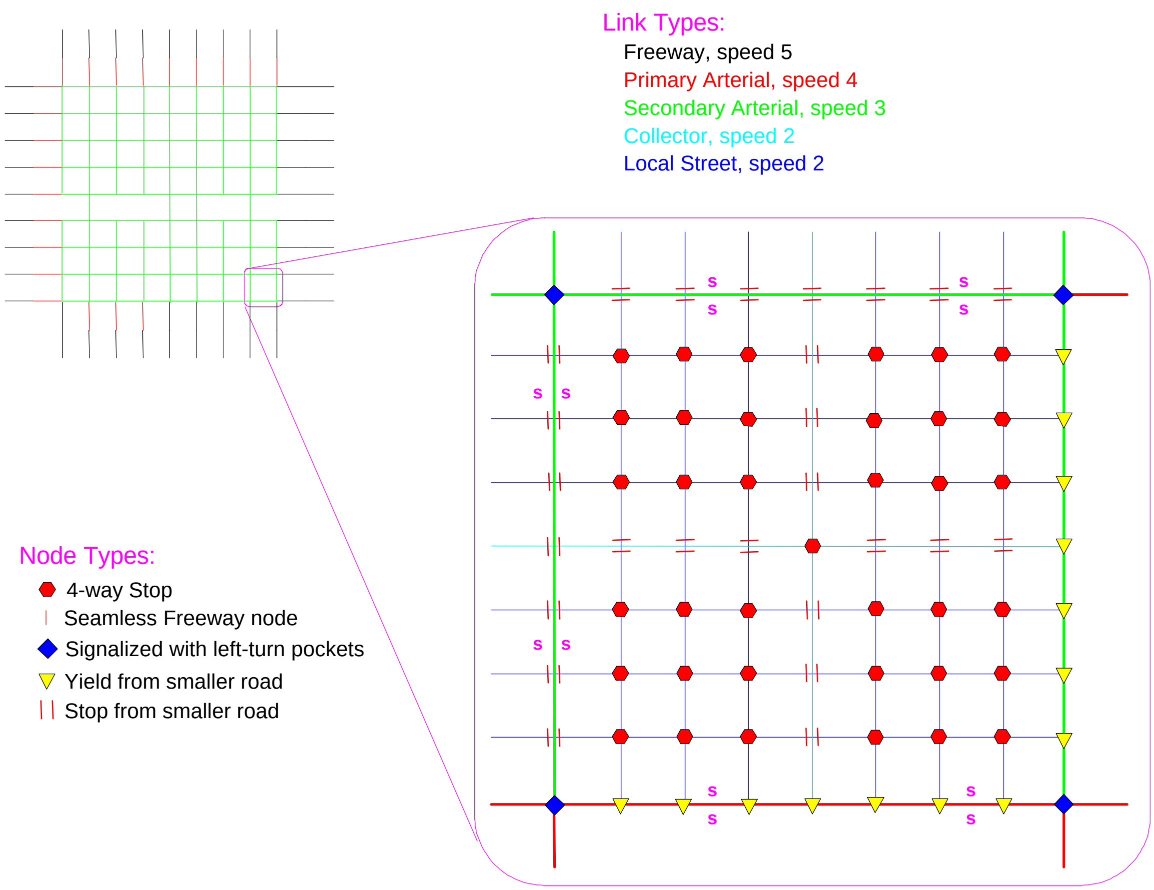


Fig. 11. The Bignet Network (S = transit stop).

Projecting out of the 8x8 grid of local streets (where the secondary arterials would have continued) are primary arterials (red) with two lanes in each direction and a speed of 30 m/s (four cells per second). Connected to these are freeways (black) with two lanes in each direction and a speed of 37.5 m/s (five cells per second).

The primary arterials are 1,000 meters long; the freeway links connected to them are 450 meters long. The node in between them has no control because the only change is in the speed limit. There are no link features (parking, activity locations, transit stops, etc.) on the primary arterials, but at the outside end of each of the freeways are one parking location and two activity locations (one on either side). All of these features are 50 meters from the outside end of the links.

The secondary arterials (green) have two lanes in each direction and a speed limit of 22.5 meters per second (three cells per second). They are all 225 meters long. (Note that a single green line on the left half of Fig. 11 has eight links per block, as shown on the right half of the figure.)

In cases in which secondary arterials intersection with local or collector streets, the secondary arterial takes precedence, whereas the smaller street must always stop or yield (yielding takes place on outside edges only).

When two secondary arterials intersect, there is a signalized intersection with left-turn pockets for each incoming link. The timing and phasing of the signals is fixed and the same for all signals throughout the network (see Fig. 12).

As shown in Fig. 12, the signal functions break down into four distinct phases:

- During phase 1, the east-west links have a protected left turn from the left-turn pocket. All links may make a right turn after an initial stop.
- During phase 2, the east-west links have a green light (protected) for the straight-through path and may make right turns without stopping (protected). The north-south links may still make a right turn after stopping. The east-west links may make a left turn, but only if there is no oncoming traffic—the links are not protected during this phase.
- Phases 3 and 4 duplicate phases 1 and 2, with the east-west directions switched with north-south.

The center of every secondary arterial has

1) one parking location accessible from both directions, and
2) one activity location on each side of the street.

At the point at which the secondary arterial forms the boundary of two zones, one activity location resides in each zone. Also in the center of the link is one transit stop in each direction, although not every secondary arterial link has a transit-stop pair.

Buses stop twice in each block group in each direction: they stop once on the third link from the west or south and once on the seventh link from the west or south. Bus routes run straight paths along all east-west secondary arterials and along all north-south secondary arterials that have bridges.

Bridges are a special case of secondary arterials. There are only two links from one side of the river to the other: a 1,575-meter link to the South, and a 225-meter link to the north. There are no stops, parking, or activity locations on the bridges. Left-turn pockets are available where the bridges intersect the secondary arterials on either side of the river. The signal timing and phasing at the locations are the same as for all other signals.

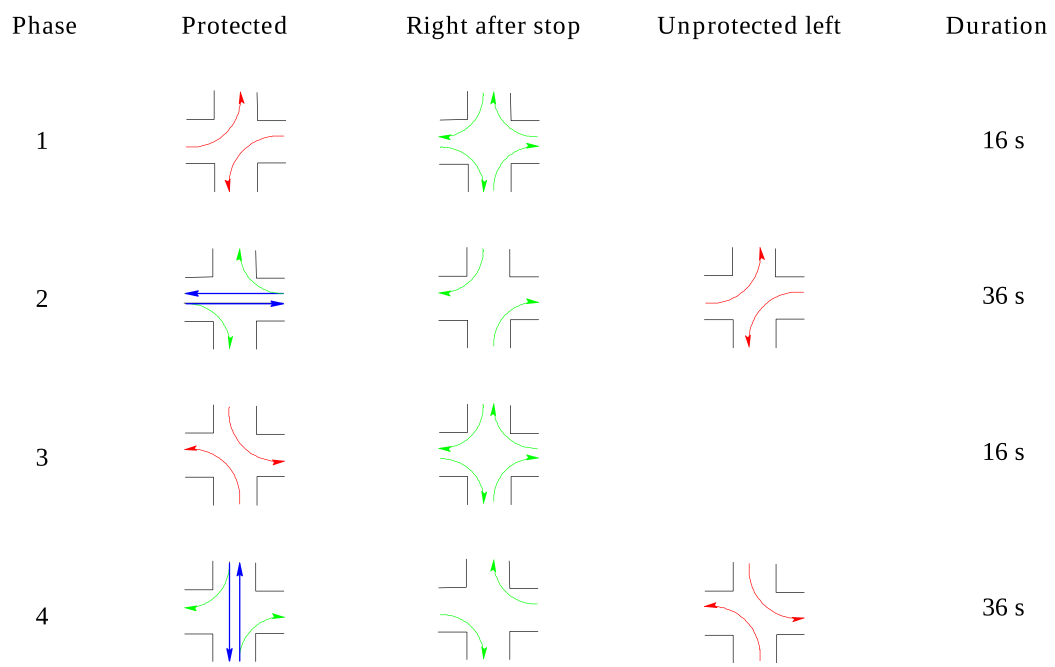


Fig. 12. Signal timing and phasing in Bignet.

Although all end-links sticking out of the network are classified as freeways, when we refer to “the freeway” in this document, we refer to the single east-west road just south of the north bank of the river. This freeway has three lanes in each direction and a maximum speed of five cells per second (37.5 m/s). There are only three places to enter the freeway: at either end of the local street grid, or on the second bridge from the west. At all three locations, there are two 75-meter merge lanes to enter the freeway in both directions, as well as two 75-meter turn lanes for exiting the freeway. The freeway has no transit stops. There also are no activity or parking areas, except at the east and west ends of the freeway (and these are outside of local streets).

Local and collector streets are considered the same except that the latter have precedence over the former where they intersect (see the right half of Fig. 2). Both streets (each 225 meters long) have one lane in each direction and a speed limit of 15 meters per second (2 cells per second). At the center of each street is one parking location accessible from both locations and one activity location on each side of the street.

Land use determines the use of activity locations, such as home, work, and shopping. A four-way stop is in place where two local or collector streets intersect. Where collector and local streets intersect, the collector has precedence and travelers on the local street must stop before continuing. Travelers advance when there is an opening in the traffic along the collector street.

Bignet’s light rail line has its own rail-only links on the north side of the river, as well as crossing the river. However, the rail travels on the secondary arterials on the south side of the river. The rail circles the downtown area then returns to the rail-only links and its origin. Both ends of the route have a parking location that functions as a yard for the trains; there also are stops distributed along the route.

On the east-west portion of the north-of-the-river part of the route, there are two stops in each direction. On the north-south portion of this part, there are five stops in each direction, increasing in frequency closer to the river. There are no stops on the bridges.

On the secondary arterial portion of the trip, the rail line stops twice in every block group in each direction: once on the third link from the west or south and once on the seventh link from the west or south—the same as busses. In this portion of the route, the rail line uses the same stops as the busses (which run on the same links).

Throughout the network, vehicles can travel anywhere except for the rail-only links north of the river. Conversely, the light rail can only use the rightmost lane while traveling on the secondary arterials around the downtown area (vehicles can use any lane). Travelers walk anywhere on the network except for the following:

- rail-only links,
- bridges, and
- freeway.

All transit vehicles may use only the links on their routes. These vehicles are scheduled to arrive at a strop every ten minutes, although the actual arrival time depends on network congestion and the number of people using the route.

2.3.1.3 Lane Connectivity

Much like the link-to-link connectivity through nodes, TRANSIMS also requires lane-to-lane connectivity. On the local street grid, the lane connectivity follows a common sense approach. From a local or collector street, travelers may turn left or right or go straight from the only lane that exists. At signalized intersections, left turns only can be made from the left turn lane and right turns only from the right lane. All of the connectivity is shown in Fig. 13.

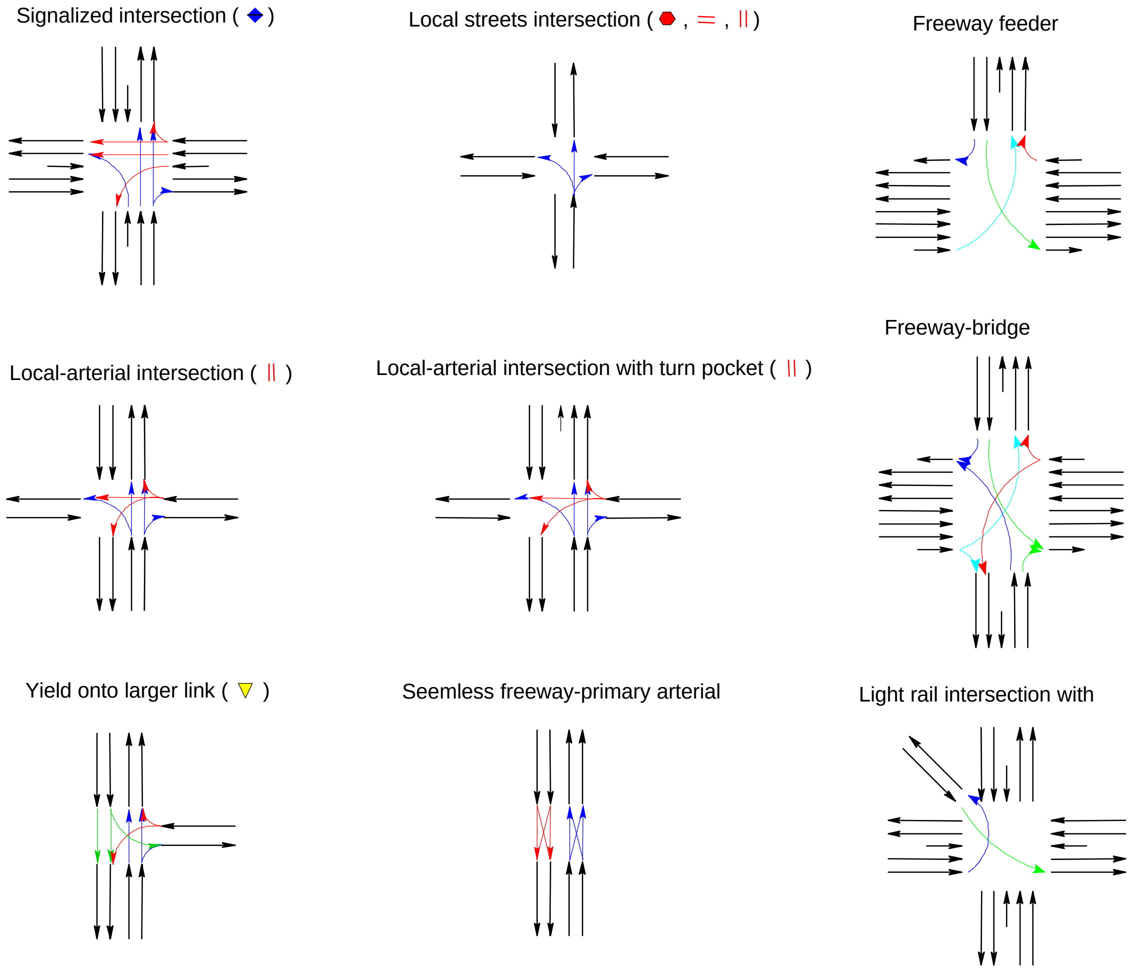


Fig. 13. Lane connectivity in Bignet.

The only part that is possibly confusing is the connectivity with the freeway in the third column. The freeway has merge lanes for entering and turn lanes for exiting. The center six lanes are accessible only from these two lanes, not from connecting links.

2.3.2 File and Directory Descriptions

The Bignet directory is located relative to the main TRANSIMS release directory, which is created during the installation process.

```
$TRANSIMS_HOME/data/bignet
```

There are two files located in this directory:

- the main run script, and
- the configuration file *bignet.cfg*.

There will be one more after the run because the main log file is written to this directory.

This folder contains two other directories: *data* and *network*. There is also a third directory that you will need to make for the resultant output. The following sections describe these directories.

2.3.2.1 Data Directory

The *data* directory contains all of the source data for Bignet. Source data does not change during the course of the run, nor is any output from any module written to the directory (such data go to the *output* directory). Files are broken down as follows:

- *data/actgen* – survey data, network zone information, and weighting factors used by the Activity Generator.
- *data/pop* – census data for the Population Generator.
- *data/transit* – plans, routes, schedules, and vehicle files for all of the transit on Bignet.

The *data* directory contains the following files:

- zone tables and time table for the trip-table activity generator,
- vehicle prototype file listing data for the vehicle types to be used in the simulation,
- list of allowed modes (walking, auto, etc.), and
- files listing the links and nodes to be included in the output files from the microsimulation.

2.3.2.2 Network Directory

The *network* directory contains the 18 tables necessary to describe the network to the TRANSIMS modules, every one of which uses these files. A description of the contents of these files can be found in Volume Two (*Networks and Vehicles*).

2.3.2.3 Output Directory

Created during the simulation run, the *output* directory houses any output by a TRANSIMS module. Some of the resultant data from one module are then used as input for another module, but the unifying feature is that the data have been generated (as output) by some TRANSIMS module at some point in the run.

After completing a run, the *output* directory will contain approximately thirty files, including population files, activities, route plans, vehicles, and all of the microsimulation output. In addition, many of these files have an index created so that the data in them can be accessed faster by later modules. When filled, this directory may contain approximately one gigabyte of data.

2.3.3 Before Running Bignet

The following sections will enable you

1) to set the necessary environmental barriers, and

2) prepare an output directory required for any TRANSIMS run.

Make sure to check the *README.txt* file for any last-minute changes not included in this documentation.

2.3.3.1 Environment Variables

The TRANSIMS executables should be in *$TRANSIMS_HOME/bin/*, with *$PVM_ROOT* serving as the base of the parallel virtual machine (PVM) tree. Both files should have the respective environment variables set, which in C-shell (Bourne shell instructions below) might look like the following, depending on the location of the TRANSIMS package:

```
% setenv TRANSIMS_HOME /usr/local/transims-1.1
% setenv PVM_ROOT $TRANSIMS_HOME/pvm/pvm3
```

Note that these paths reflect the location of where the TRANSIMS package was installed (see Volume Six (*Installation*)). Several other PVM environment variables also must be set:

```
% set path=($path $PVM_ROOT $PVM_ROOT/lib $PVM_ROOT/bin)
% setenv PVM_ARCH `pvmgetarch`
```

Note: Make sure that you use the left single quote in the second line above, then `PVM_ARCH` automatically will be set for Linux, Solaris, or any other architecture. This line makes sure that the path of PVM is correct.

The above instructions are for *csh* or *tcsh* command interpreters. For *sh* or *bash* interpreters (type "`echo $SHELL`" to find out), then do the following:

```
% export TRANSIMS_HOME=/usr/local/transims-1.1
% export PVM_ROOT=$TRANSIMS_HOME/pvm/pvm3
% set path=($path $PVM_ROOT $PVM_ROOT/lib $PVM_ROOT/bin)
% export PVM_ARCH=`pvmgetarch`
```

2.3.3.2 Disk space

Before running this scenario, select a location for your output directory. Remember that Bignet may require more than one gigabyte of storage (use the Unix command "`df-k`" to determine current disk usage). You may elect to assign a separate drive dedicated to TRANSIMS output, soft-linking the local output directories in the run directory to other directories on that large storage device.

Example:

```
% mkdir /mnt/bigdisk/TRANSIMS
% mkdir /mnt/bigdisk/TRANSIMS/bignet
% cd $TRANSIMS_HOME/data/bignet
% ln -s /mnt/bigdisk/TRANSIMS/bignet output
```

The *output* directory (found in *TRANSIMS_HOME/data/bignet*) will store data on the device you assign. If you know you have more than one gigabyte of free space in the directory in which TRANSIMS was installed (*TRANSIMS_HOME*), then create an output folder using the following commands.

```
% cd $TRANSIMS_HOME/data/bignet
% mkdir output
```

2.3.3.3 Modifying the Configuration File

The following steps are designed to ensure that the configuration file reflects your computer's particular configuration:

Step One While still in the Bignet directory, use an editor to open the configuration file *bignet.cfg*.

- The first line has the key `TRANSIMS_ROOT` on it.
- Make sure the value of this key reflects the location on your system of the TRANSIMS package (it should already be correct from the installation process).

Step Two Search for the string "`PAR_HOST_0`."

- This string specifies the name of the machine on which you will be running the microsimulation.
- Edit its value to have the fully qualified host name.
  Note: only the PVM version of the microsimulation needs this key set.

Step Three On the next line (`PAR_SLAVES`) is the number of CPUs the microsimulation will use.

- Enter an integer less than or equal to the total number of CPUs in the machine, but greater than or equal to one.

Step Four Search for `ROUTER_NUMBER_THREADS` and set this equal to the same number used in `PAR_SLAVES`.

- This will be the number of CPUs used by the Route Planner.

For both the Route Planner and the Traffic Microsimulator, note that you can "lie" to the programs and tell them to use as many or as few slaves and threads as you want. The above instructions will produce the optimal run.

Once you have completed this process, you are ready to run TRANSIMS on the Bignet Network.

2.3.4 Using the Script to Run Bignet

During your first simulation run, we advise using the provided shell script. If you only want the output and are not interested in a tutorial on a complete run of TRANSIMS, then this is all you will need.

To set the simulation in motion, run the following script in the Bignet directory: *main_run.script*. This script does everything, from checking the environment variables to generating the population to running the Emissions Estimator to preparing the resultant output for the Output Visualizer. The following steps will enable you implement this script:

- Go to the Bignet directory:
  `% cd $TRANSIMS_HOME/data/bignet`

- Run the script in a C-shell command interpreter by using
  `( nohup time main_run.script ) >&! main.log &`

- or in a Bourne shell command interpreter using
  `( nohup time main_run.script ) >& main.log &`

The `nohup` command enables you to logout without "killing" the process, the last "&" returns control of the terminal to you, and the `time` command records a total runtime at the end of the main log file *main.log*, to which is written both standard output and standard error (by way of the "`>&!`").

If other users will be using the system and you want to be polite, insert a `nice` command between the `nohup` and the `time` commands. Because you are running this with a `nohup` command, feel free to completely log out.

Warning: Your system administrator may (1) have processes automatically killed when you log out, or (2) set a maximum system usage for you. In any case, you will want to talk to that individual before running Bignet.

2.3.4.1 Keeping Track of the Script's Progress

To check a run's progress, you can use the log files that Bignet creates. The main log file, *main.log*, records only the time-consuming components of the run, as well as the time it took to run them. If you check while it is running the Traffic Microsimulator (CA, for cellular automata), it might look like the following (use "*cat main.log*"):

```
Running the Population Synthesizer ...
1495.0u 196.0s 28:38 98% 0+0k 0+0io 0pf+0w
Running the Activity Generator ...
1333.0u 4.0s 22:18 99% 0+0k 0+0io 0pf+0w
Running the Router ...
61446.0u 210.0s 17:09:05 99% 0+0k 0+0io 0pf+0w
Running the CA ...
```

You can look at the "`man`" page for the time command by using the `man time` command, but do so briefly. The third field is the run time in hh:mm:ss format (hours:minutes:seconds). In the above example, the Population Synthesizer took 28 minutes to run, the Activity Generator 22 minutes, and the Route Planner 17 hours. The Traffic Microsimulator (CA) was still running, so there is no time entry yet.
If this file has any error messages in it, the run probably failed and you will need to learn more about TRANSIMS to troubleshoot the problem.

There are also log files for each individual TRANSIMS module. Some of these would have gone to standard output but were redirected to a log file (as in the main script run command above). The other log files created by redirecting the standard output are as follows:

- *Syn.log*,
- *BlockGroupLoc.log*,
- *Vehgen.log*,
- *ActivityGenerator.log*,
- *ActTripGen.log*,
- *PopConverter.log*,
- *Router.log*, *pvm.log*,
- *CA.log*,
- *indexvehtobin.log*,
- *ConvertVELfile.log*, and
- *EmissionsEstimator.log*.

Some of the data found in these files requires a thorough understanding of TRANSIMS. Note that *BlockGroupLoc.log* will be filled with supposed errors from households that we never intended to place on this network. This is normal, unless you are using real census data on a real network. All of these log files are located in the *output/logs* folder. Other log files that are created are as follows:

- The name of the file is a configuration file key in the configuration file *bignet.cfg*. In the released configuration file, these files all go to the output subdirectory of the main Bignet directory.

- The Route Planner creates a *router.problems* file that lists information about all of the travelers who could not be routed.
- The Traffic Microsimulator creates a logfile that supplements what is in *CA.log*.
- Looking at the end of the logfile file will tell you how far the microsimulation has progressed (use the `tail -20 logfile` command to see the last twenty lines). You can check the total time being simulated in the configuration file *bignet.cfg* by looking for the `CA_SIM_START_*` keys and the `CA_SIM_STEPS` key.

Note that many log files, particularly the *router.problems* file and CA logfile, will be filled with errors that normally would be fixed using feedback, which we are not using in this example.

2.3.5 Running Bignet Manually

If you have some knowledge of *csh* script files, you can use the one provided in this release (see Section 1.1.4, Using the Script to Run Bignet). Simply cut and past the commands described in this section. Note that the one difference between this run and the one done by the script is that, in this case, all standard output log files are created in the run directory. See the script if you want them in output/logs.

As above, first be sure you have all of the environment variables set, then go to the Bignet directory

```
% cd $TRANSIMS_HOME/data/bignet
% set RUNDIR = `pwd`
```

This second line sets a variable for later use (thus avoiding additional typing).

2.3.5.1 Make a Population

Creating a population involves three steps:

1) create the households from census data,
2) locate the households in appropriate block groups, and
3) assign vehicles to the households.

Because the Population Synthesizer's output is not used directly by other modules, we put it in a separate directory. Create the directory for Population Synthesizer output and a temp directory (also for the Population Synthesizer) in the Bignet output folder. Change to the former directory as follows:

```
% mkdir output/synpop
% mkdir output/tmp
% cd output/synpop
```

Now run the Population Synthesizer:

```
     % TRANSIMS_HOME/bin/Syn $RUNDIR/bignet.cfg >&!
$RUNDIR/Syn.log
```

Running the Population Synthesizer may take 30 minutes on a 300-MHz machine. Because no progress report is written to the log file, you will have to assume that it is running correctly. You also can check the size of the files in the *tmp* folder you created above to see that they are changing. They can be deleted when the run is complete.

The Population Synthesizer makes separate files for family households, non-family households, and group quarters. Before running the population locator, use the *Syn* program to concatenate all the created files. To do this, open both non-family files (in the present working directory), *Base_Non_Family_Synthetic_HHRecs.out* and *Base_Group_Synthetic_HHRecs.out*, in an editor and remove the first two lines of each. Once this is done, run the following:

```
     % cat Base_*_Synthetic_HHRecs.out >! ../population.baseline
```

or by doing something completely on the command line, such as

```
% cp Base_Family_Synthetic_HHRecs.out ../population.baseline
% gawk 'NR>2{print}' Base_Non_Family_Synthetic_HHRecs.out >>
../population.baseline
% gawk 'NR>2{print}' Base_Group_Synthetic_HHRecs.out >>
../population.baseline
```

Note: It is important that you use the right single quote with the `gawk` command. Once you are sure of this, change back to the main run directory and run both the population locator and the vehicle generator to complete this step in the run:

```
% cd $RUNDIR
% $TRANSIMS_HOME/bin/BlockGroupLoc bignet.cfg >&!
BlockGroupLoc.log
% $TRANSIMS_HOME/bin/Vehgen bignet.cfg >&! Vehgen.log
```

Note that *BlockGroupLoc.log* will report certain errors that you can safely ignore. The file has 409 Tract/BlockGroup pairs that do not get located because we are not using all of the block groups in PUMA 1300 of the census data we selected. This generates approximately 160,000 error messages.

2.3.5.2 Generate Activities for the Located Population

To generate activities, use the following steps:

1) Convert the format of the located population file, since the Activity Generator does not use the data format from the census population:

```
% $TRANSIMS_HOME/bin/PopConverter bignet.cfg >&! PopConverter.log
```

2) Run the activity generator for the located population

```
% $TRANSIMS_HOME/bin/ActivityGenerator bignet.cfg >&! ActivityGenerator.log
```

Both of these commands create log files that primarily contain error messages. Because Bignet does not use “real” data on a “real” network, some of the Portland survey activities will not fit the Bignet population. Thus, there will be approximately 11,000 warnings and errors in the *ActivityGenerator.log* file. This is to be expected. (These errors could be easily corrected using feedback.) The Activity Generator may take approximately 25 minutes on a 300-MHz machine.

2.3.5.3 Create Itinerant Travelers and Freight Using Trip Tables

Not all travelers in a municipality reside there. TRANSIMS simulates these non-resident individuals by using origin-destination trip tables. Freight activities are also generated using origin-destination trip tables.

The *ActTripGen* program creates the population, vehicle, and activity files for these people. This program is complicated to use because it can only do one vehicle type at a time and has only one set of configuration file keys. To remedy some of this complexity, you can create a new directory in the output folder and make a copy of the configuration file:

```
% mkdir output/trips
% cp bignet.cfg bignet.trips.cfg
```

Once this is done, you must edit the new configuration file for each of the five runs of the *ActTripGen* program. This can be done in an editor by changing the appropriate line in the *new* configuration file or by adding a new line to the end of the new configuration file (because only the last occurrence of each key is used in TRANSIMS programs).

For example, if you are instructed to set the `VEH_VEHICLE_TYPE 1` configuration file key, you could either search the configuration file *bignet.trips.cfg* and change the entry (so long as there is only one), or you could add that line to the end of the file in the editor. Both have the same result. (There is a third command-line-only option demonstrated in the run script; check there to see it.) For information on what each of these keys does and why we're changing them, see Volume Three (*Modules*), Chapter Two (*Population Synthesizer*), Section 7 on the Trip Table Activity Generator program.

The first trip table travelers are the itinerant travelers, and we add 3,000 of them as specified in the zone and timetables. To do this, set the following keys

```
VEH_VEHICLE_TYPE 1
ACT_TRIPTABLE_STARTING_HH_ID 6000
ACT_TRIPTABLE_STARTING_PERSON_ID 6000
ACT_TRIPTABLE_STARTING_VEHICLE_ID 6000
ACT_TRIP_TABLE_OUTPUT
$TRANSIMS_ROOT/data/bignet/output/trips/activities.trips.1
TRIP_TABLE_VEHICLE_FILE
$TRANSIMS_ROOT/data/bignet/output/trips/vehicles.trips.1
POP_TRIPTABLE_FILE
$TRANSIMS_ROOT/data/bignet/output/trips/population.trips.1
```

```
ACT_TRIPTABLE_FILE
$TRANSIMS_ROOT/data/bignet/data/trip.zonetable1
```

Once this is done, run the program using

```
     % $TRANSIMS_HOME/bin/ActTripGen bignet.trips.cfg >!
ActTripGen.log
```

Note: make sure that you edit the new trip table configuration file each time and that you are running it with that file.

The next four groups of trip table travelers are the freight trips. We split these among the four truck types present in this release. The first truck group, light trucks, is created by setting the following configuration file keys:

```
VEH_VEHICLE_TYPE 2
VEH_VEHICLE_SUBTYPE 1
ACT_TRIPTABLE_STARTING_HH_ID 9001
ACT_TRIPTABLE_STARTING_PERSON_ID 9001
ACT_TRIPTABLE_STARTING_VEHICLE_ID 9001
ACT_TRIP_TABLE_OUTPUT
$TRANSIMS_ROOT/data/bignet/output/trips/activities.trips.2
TRIP_TABLE_VEHICLE_FILE
$TRANSIMS_ROOT/data/bignet/output/trips/vehicles.trips.2
POP_TRIPTABLE_FILE
$TRANSIMS_ROOT/data/bignet/output/trips/population.trips.2
ACT_TRIPTABLE_FILE
$TRANSIMS_ROOT/data/bignet/data/trip.zonetable2
```

Once this is done, run the trip-table activity generator using

```
     % $TRANSIMS_HOME/bin/ActTripGen bignet.trips.cfg >>
ActTripGen.log
```

which will append to the already existing log file. The above line is the same one you will use for each run of this program below, so it will not be repeated.

The second group of trucks is created with the following keys and a run of *ActTripGen*:

```
VEH_VEHICLE_SUBTYPE 2
ACT_TRIPTABLE_STARTING_HH_ID 10000
ACT_TRIPTABLE_STARTING_PERSON_ID 10000
ACT_TRIPTABLE_STARTING_VEHICLE_ID 10000
ACT_TRIP_TABLE_OUTPUT
$TRANSIMS_ROOT/data/bignet/output/trips/activities.trips.3
TRIP_TABLE_VEHICLE_FILE
$TRANSIMS_ROOT/data/bignet/output/trips/vehicles.trips.3
POP_TRIPTABLE_FILE
$TRANSIMS_ROOT/data/bignet/output/trips/population.trips.3
```

To create the third set of trucks, set the following keys and run the program (see command line above):

```
VEH_VEHICLE_SUBTYPE 3
ACT_TRIPTABLE_STARTING_HH_ID 11000
ACT_TRIPTABLE_STARTING_PERSON_ID 11000
ACT_TRIPTABLE_STARTING_VEHICLE_ID 11000
ACT_TRIP_TABLE_OUTPUT
$TRANSIMS_ROOT/data/bignet/output/trips/activities.trips.4
TRIP_TABLE_VEHICLE_FILE
$TRANSIMS_ROOT/data/bignet/output/trips/vehicles.trips.4
POP_TRIPTABLE_FILE
$TRANSIMS_ROOT/data/bignet/output/trips/population.trips.4
```

The fourth truck type (the largest) is implemented with the following keys:

```
VEH_VEHICLE_SUBTYPE 4
ACT_TRIPTABLE_STARTING_HH_ID 12000
ACT_TRIPTABLE_STARTING_PERSON_ID 12000
ACT_TRIPTABLE_STARTING_VEHICLE_ID 12000
ACT_TRIP_TABLE_OUTPUT
$TRANSIMS_ROOT/data/bignet/output/trips/activities.trips.5
TRIP_TABLE_VEHICLE_FILE
$TRANSIMS_ROOT/data/bignet/output/trips/vehicles.trips.5
POP_TRIPTABLE_FILE
$TRANSIMS_ROOT/data/bignet/output/trips/population.trips.5
```

As in each case, run of the *ActTripGen* program once this is done.

Once you have completed the truck types and have the program running, the next step is to combine these different files into

1) a single activity file, and

2) a single vehicle file.

Nothing need be done with the population files because they will not be used. The activity files are easier because they have no header.

As a first step, go to the directory containing the data and concatenate them into a file in the output directory:

```
      % cd output/trips
      c% at activities.trips.? >! ../activities.trips
```

For the vehicle files, you will need to open those not ending in “*.1*” in an editor and remove the header line. Then you can run the following:

```
      % cat vehicles.trips.1 vehicles.trips.2 vehicles.trips.3
>! ../vehicles.trips
      % cat vehicles.trips.4 vehicles.trips.5 >>
../vehicles.trips
```

Finally, return to the run directory and move that trip configuration file out of the way

```
% cd $RUNDIR
% mv bignet.trips.cfg output/trips/
```

2.3.5.4 Route all travelers

Before running the Route Planner, combine all of its input files so there is only one of each type. First combine the transit schedules and routes for both busses and the rail line

```
     % cat data/transit/bus_routes data/transit/rail_route >!
data/transit.routes
     % cat data/transit/bus_schedules data/transit/rail_schedule
>! data/transit.schedules
```

Note that while transit drivers are not routed (they have pre-set plans), the transit schedules are required by the Route Planner to get passengers on transit.

Once you have completed the above step, combine the vehicle files (this demonstrates the way to concatenate files and remove headers without using an editor):

```
% mv output/vehicles output/vehicles.people
% cp data/transit/bus_vehicles output/vehicles
% gawk 'NR!=1{print}' data/transit/rail_vehicles >>
output/vehicles
% gawk 'NR!=1{print}' output/vehicles.trips >> output/vehicles
% gawk 'NR!=1{print}' output/vehicles.people >> output/vehicles
```

Once this is done, combine the activity files

```
% cp output/activities output/activities.people
% cat output/activities.trips >> output/activities
```

At this point, the Route Planner is ready to run. The execution time on a fast parallel machine (one with 12 processors at 300 MHz) may be less then one hour. Slower machines will take much longer (perhaps more than one day) and will use approximately 400 megabytes of RAM.

To run the Route Planner, use the following:

```
% $TRANSIMS_HOME/bin/Router bignet.cfg >&! Router.log
```

The log file for the Route Planner will keep track of

1) how many travelers have been routed, and

2) errors.

Once the Route Planner run is completed, you will need to prepare for the microsimulation. The first step is to merge the plans by using the program *PlanFilter* for the transit drivers with those of the passengers and other travelers. To do this, execute the following:

```
% mv output/plans output/plans.people
```

```
% $TRANSIMS_HOME/bin/PlanFilter -d -o output/plans
data/transit/bus_plans data/transit/rail_plans
output/plans.people
```

This completes the third major step in running TRANSIMS.

2.3.5.5 Run the Traffic Microsimulator (CA)

This is the final step in simulating the Bignet community. For details regarding how to run the CA, Volume Three (*Modules*), Chapter Five (*Traffic Microsimulator*).

Before starting this run, you first must obtain the PVM, which is used to handle the parallel communication of the processes in the traffic microsimulation, setup and run. First, check your *rhosts* file to make sure there is an entry for the present machine

```
% cat ~/.rhosts
% echo $HOST $USER
```

If the result of the second command is not an entry in the result of the first, or if the *rhosts* file does not exist, then you need to create the file using the following:

```
% echo "$HOST $USER" >! ~/.rhosts
% chmod 600 ~/.rhosts
```

The second line above closes a potential security hole.

At this point it is possible to start the PVM. To begin, type “`pvm`” at the command line then type “`quit`” at the PVM prompt. It should then display a message that says “pvmd still running” and that you are OK.

While the traffic microsimulation may be the longest running program in the TRANSIMS framework, starting it is quite simple:

```
% $TRANSIMS_HOME/bin/ARCH.PVM.$PVM_ARCH/CA bignet.cfg >&!
CA.log
```

Depending on your machine, this run may take an hour or a few days. You can check on its progress by looking at the end of the log file called *logfile* in the output directory and comparing it to the specified stop time in the configuration file. When it eventually finishes, you can stop the PVM daemon by typing “`pvm`” at the command line then type “`halt`” at the PVM prompt.

The basic run of TRANSIMS is now complete. It is now time to examine the results.

2.3.5.6 Postprocessing

Postprocessing the resultant data from the Traffic Microsimulator enables the user to make it easier (and sometimes possible) to visualize the results. The first step in postprocessing is to convert the ASCII vehicle snapshot output into a binary file. Such a conversion enables the Output Visualizer to read it much faster. To execute this conversion, use the following command line:

```
      $TRANSIMS_HOME/bin/indexvehtobin output/snapshot.veh
output/snapshot.veh.bin >&! indexvehtobin.log
```

The conversion process takes a few minutes. Once it is done, prepare the Emissions Estimator data for analysis. Note that this program takes the file to be processed as an argument on the command line in addition to the configuration file:

```
      % $TRANSIMS_HOME/bin/ConvertVELfile
output/emission_summary.vel bignet.cfg >&! ConvertVELfile.log
```

The next step is to perform the emissions estimation in the output directory (because it write to the present working directory):

```
      % cd output
% $TRANSIMS_HOME/bin/EmissionsEstimator $RUNDIR/bignet.cfg >&! %
$RUNDIR/EmissionsEstimator.log
```

This process may take several hours to run.

The final step is to postprocess summary data. This data will show you the average density and velocity on links in 15-minute bins. The TRANSIMS-executable `SpaceSumFilter` will accomplish the output data filtering. It will filter the file named in the configuration file key `OUT_SUMMARY_NAME_1` with the *.spa* extension, which is created if the value of the key `OUT_SUMMARY_TYPE_1` includes `DENSITY`. (The supplied configuration file will do this.) The filtered file name is the second argument to the function:

```
      % cd $RUNDIR
      % $TRANSIMS_HOME/bin/SpaceSumFilter bignet.cfg
output/summary.spa.filtered
```

Filtering may take several minutes.

There are other data files that would need some postprocessing before they can be visualized, but we stop our example at these. To see how to convert all the other forms of data for visualization, see Volume Three (*Modules*), Chapter Eight (*Output Visualizer*).

2.3.6 Visualizing the Bignet Output

At the time of this release, it is difficult to use the Output Visualizer to examine the output from the Population Generator or the Activity Generator. For these modules you must page through the output files and the chapters that describe them.

The route plans can be converted for visualization, but in this section we only demonstrate the three types of output postprocessed in the preceding section. We will also look through one of the event files using a filter that converts data into English.

In the Bignet directory, start the Output Visualizer with the following:

```
% cd $TRANSIMS_HOME/data/bignet
% $TRANSIMS_HOME/bin/Vis bignet.cfg
```

This operation may take a few minutes to load the network. Once completed, select File→Load Indexed Vehicles from the menu at the top. This selection brings up a file-selection box in which you check on the output/ folder, single click on the *snapshot.veh.bin* file, and click [OK].

At this point the vehicle snapshots recorded from the Bignet run will begin to load. This operation may take several minutes to complete. Moreover, it may require more than 350 megabytes of RAM. Therefore, if you have less that that on your machine, the operation may cause a system crash.

Once the operation is complete, select View→Vehicles, which will bring up a dialog box. Select "View by Type" in this box, after which you should change the size to approximately 1/3 of full by using the bottom slider. Once this is done, select the "3D Vehicles" option and click [OK].

To start the time sequence, left click on the display. Each snapshot represents ten simulated minutes. Notice the time displayed in the lower left.

At the top-right of the screen there are zoom and rotation sliders that enable you to make fine adjustments. In addition, the view-vehicles selections can be colored differently, with by-type and by-passenger proving to be the most useful options. (If you select "color by velocity," note that you may need to change the threshold slider at the bottom of the main window to see the vehicles; it always starts at the wrong end for this type.)

2.3.6.1 Emissions Output

To clear the vehicle data from memory, use Select File→Close Indexed Vehicles. Another option is to quit that Output Visualizer and start again (this is done because the Visualizer sometimes becomes "confused" when closing data and opening new data sets).

Upon using Select File→Load Variable-Size Box Data, a file dialog box will appear. Single click on the *output/* folder, single click on the *emissions.out* file, and click [OK]. The emissions data may take some time to load.

Before animating the data, select View→Variable-Size Boxes, which will display

- at the top, a list of variables that are available from the missions data, and
- in the middle, a series of display options.

At this point, set the 3D-scale factor equal to 2.0, then click [OK]. Once this is done, select View→Legend, which displays a legend on the right of the plotting area. To make the data easier to see when you zoom in, use Select Modes→Lights On/Off.

To animate, click the forward arrow button on the lower right. It will advance the display to the next time bin, which includes morning rush-hour activity. The variable being displayed is both below the legend in black and to the right of the time on the lower left. When you first arrive here, the variable should read "VTT," which consists of the average velocity in a box.

The size of boxes for all of the emissions data is 30 meters long. There are boxes for each direction on each link. To view the next piece of emissions data, the nitrous oxide (NOX), select View→Increment Column. Continue incrementing the column and looking at the time bins. In addition, try zooming in on an interesting part of the network. To learn more about the emissions data, see Volume Three (*Modules*), Chapter Seven (*Output Visualizer*).

2.3.6.2 Non-emission Summary Data

Once you close down the variable-size box data (under the "File" menu) or restart the Output Visualizer, initiate Select Edit→Change All Colormaps and select the file *data/VisSummaryMaps.bin*. This series of steps ensures that low densities appear as green and that high densities appear as res. It also ensures that slow speeds display as red and high speeds as green. (The configuration file selects the colormap for emissions by default.)

In the Select File→Open Variable-Size Box Data, find the filtered output file created during postprocessing (see Section 1.1.5.6). In Select View→Variable-Size Boxes, set the 3D-scale factor to 100,000. Set the "Colormap Max Val" field to approximately 0.2. (You may want to select Edit→Background Color. A medium gray background (with all sliders at approximately 140) ensures that all colors used to paint boxes in the default color map are different from the background.)

To start animating, first press the left mouse button, which activates a display that shows boxes whose color and height are determined by the average density over a 15-minute period. (You may elect to select Modes→Overlay On/Off for faster animation, but this option sometimes makes the links disappear.)

In Select View→Variable Size Boxes, set the colormap column to 1. When this is done, the display shows boxes whose height is proportional to the average density as before, but whose color is determined by the average speed in the box over a 15-minute interval.

2.3.6.3 Additional Output Visualizer Features

The Output Visualizer has many other features, including the ability to look at plans and any type of summary data. Almost any type of output data can be converted to a form the Output Visualizer can understand. To learn more about such conversions, see Volume Three (*Modules*), Chapter Eight (*Output Visualizer*).

2.3.7 Event Output Collected from the CA

The event output files record all major events in the simulation, although in this example they have been filtered into several files. To locate these files, go to the output directory and you will find the following:

- *event.lost.trv* – records every occurrence of a traveler getting lost.
- *event.anomaly.trv* – records all other anomalies.
- *event.trips.trv* – contains a record of the beginning and the end of every trip in the simulation.
- *event.endsim.trv* – records any end-of-simulation activity, which could include travelers still on the network (who therefore would have an odd number of entries in the *trips* event file).

There is one more event file, which records all events for all individuals in household 24507, *event.hh_24507.trv*. Because these files use a compounded status flag, we provide a script that takes the file as an argument. For example, it can be run with the following:

```
    % cd $RUNDIR/output/
    % gawk -f $RUNDIR/scripts/event.awk event.hh_24507.trv |
more
```

The `more` command at the end of the second line will show you only one screen of data at a time. The following keyboard functions provide navigation:

- Press the space bar to advance to the next page.
- Press <`b`> to return to a previous screen.
- Press <`q`> to quit.

This data could also be sent to a file (for example, replace "`| more`" with "`>! event.hh_24507.txt`"). Try this with some of the other files as well. Note that some of the event files are missing some fields (the script will tell you), so some of the fields in the output may be blank for those event files. Although this script should work for any TRANSIMS event file in this release, it has been tested only on Bignet.

3. TUTORIALS

3.1 Population Synthesizer

3.1.1 Running the Population Synthesizer

Before running the Population Synthesizer, the environmental TRANSIMS_HOME variable must be set (see Troubleshooting). Once this is done, perform the following four steps.

Step One Create a working directory in which the Population Synthesizer's output files will be generated.

Step Two Change directory (cd) into the working directory.

Step Three Copy the following configuration file into the working directory.

```
$TRANSIMS_HOME/data/synpop/sample/example.cfg
```

Edit the configuration file and add a line at the top of the file to define the configuration file key TRANSIMS_ROOT to be the full pathname of the base directory of your TRANSIMS installation (TRANSIMS_HOME).
Example:

```
TRANSIMS_ROOT /disks/clarisse1/v1.1
```

Change all references to TRANSIMS_HOME in the following lines to be TRANSIMS_ROOT.
Example:

```
SYNPOP_BASE_DIRECTORY        $TRANSIMS_ROOT
SYNPOP_STF_INFO_DIRECTORY    $TRANSIMS_ROOT/data/synpop/Parep2/stf
SYNPOP_STF_DATA_DIRECTORY    $TRANSIMS_ROOT/data/synpop/sample/stf
SYNPOP_MABLE_FILE            $TRANSIMS_ROOT/data/synpop/sample/mable/geocorr.csv
SYNPOP_PUMS_DIRECTORY        $TRANSIMS_ROOT/data/synpop/sample/pums
SYNPOP_MARGINALS_FILE        $TRANSIMS_ROOT/data/synpop/sample/marginals.txt
```

You may want to edit the configuration file to change the output file name prefixes for the base of forecast populations. To do this, use the following configuration file keys:

SYNPOP_BASE_PREFIX and SYNPOP_FORECAST_PREFIX

You may want to change the output file name prefixes for the household or person demographics as well. To do this, use the following configuration file keys:

SYNPOP_HOUSEHOLD_DEMOGRAPHICS and
SYNPOP_PERSON_DEMOGRAPHICS

Appendix B lists the valid person and household demographic field names.

The Activity Generator requires the following household demographics:
- `R18UNDR`,
- `RWRKR89`, and
- `RHHINC`

The Activity Generator also requires the following person demographics:
- `AGE`,
- `RELAT1`,
- `SEX`, and
- `WORK89`

To suppress the generation of a base- or forecast-year population, remove the following configuration file key:

`SYNPOP_BASE_PREFIX` or `SYNPOP_FORECAST_PREFIX`

Step Four Once steps 1 through 3 are completed, type the executable name followed by the configuration file key `Syn example.cfg.` Note that the directory in which `Syn` is installed must be in the user's path (`$TRANSIMS_HOME/bin/Syn`).

Various information messages will be printed on the console. The Population Synthesizer takes approximately one-half hour to run this example on typical computing platforms.

3.1.2 Files

After the Synthetic Population Generator has run, four new files will be present in the working directory:

1) *user_supplied_base_prefix_Family_Synthetic_HHRecs.out*

2) *user_supplied_base_prefix_Non_Family_Synthetic_HHRecs.out*

3) *user_supplied_base_prefix_Group_Synthetic_HHRecs.out*

4) *user_supplied_forecast_prefix_Synthetic_HHRecs.out*

The first three files contain the base-year (1990) populations for the demographic types; the last file contains the forecast-year (1994) population.

3.1.2.1 File Format

The file format consists of two lines containing the names of the household and the demographic data, followed by the synthetic household data.

The first line of data consists of the following:

- tract ID,
- block group ID,
- an "H" to indicate that it is a household record,
- household ID (always –1),
- PERSONS field,
- AUTOS field,
- home location (always –1), and
- demographic data in the order listed in line one of the file (`PUMSHH, R18UNDR, RWRK89, RHHINC` in this example).

See Volume Three (*Modules*) for more information on file formats. This household record is followed by person records (one per person in the household files; group quarter records have one person per household). A person consists of the following:

- household ID (always –1),
- a "P" to indicate that it is a person record,
- person ID (always –1), and
- demographic data in the order listed in line two of the file (`AGE, RELAT1, SEX, WORK89` in this example).

3.1.3 Troubleshooting

3.1.3.1 Environmental Variables

Before running the Population Synthesizer, make sure that the environmental variables are set appropriately.

TRANSIMS_HOME – root directory of the TRANSIMS distribution. The following directory must be present and have read permission by the user:

$TRANSIMS_HOME/data/synpop

3.1.3.2 Configuration File Keys

Before running the Population Synthesizer, make sure that the following four configuration file keys are set appropriately.

1) Set SYNPOP_STF_INFO_DIR to *$TRANSIMS_HOME/data/synpop/Parep2/stf*. This directory must be present and have read permission by the user.

2) Set `SYNPOP_STF_DATA_DIR` to the directory in which STF3A data resides. Sample STF3A data is in *$TRANSIMS_HOME/data/synpop/sample/stf*. If using Oregon PUMA 01300, the following should be set to this directory: `SYNPOP_STF_DATA_DIR`. If using data from a mounted CD-ROM, set `SYNPOP_STF_DATA_DIR` to the CDROM directory (Example: */mnt/cdrom*). The directory must be available and have read permission by the user.

3) Set `SYNPOP_PUMS_DATA_DIR` to the directory in which the PUMS data resides. Sample PUMS data for Oregon is in *$TRANSIMS_HOME/data/synpop/sample/pums*. If using Oregon PUMA 01300, `SYNPOP_PUMS_DATA_DIR` should be set to this directory. If using data from a mounted CD-ROM, set `SYNPOP_PUMS_DATA_DIR` to the CDROM directory (example: */mnt/cdrom*). The directory must be available and have read permission by the user.

4) Set `SYNPOP_MABLE_FILE` to where the downloaded MABLE/Geocorr data resides. Sample MABLE data for Oregon is in the following directory:

   $TRANSIMS_HOME/data/synpop/sample/mable/geocorr.csv

   If using Oregon PUMS 01300, `SYNPOP_MABLE_FILE` should be set to this directory.

The following directory also must be available and have read permission by the user: *$TRANSIMS_HOME/data/synpop/doc*

Each PUMA should be processed separately. The PUMA ID that is entered must match <u>exactly</u> the PUMA number in the MABLE/GEOCORR file and the PUMS data (i.e., 01300, not 1300).

3.1.4 Locating a Population on a Transportation Network

3.1.4.1 BlockGroupLoc

This program generates home locations for populations on a transportation network by correlating census tract and block-group user data values specified in the network activity location file with tract and block group data in the baseline population. Candidate home locations must have the same census tract and block group as the household; they also must have residential land-use values greater than zero.

BlockGroupLoc also generates household and person IDs and assigns them to the located population. The user data in the activity location table in the TRANSIMS transportation network must contain tract, block group, and residential and commercial land-use values.

Alternative tract and block groups may be specified for households in block groups that do not have activity locations associated with the household's tract/block group. The alternative tract/block group pairs are specified in a Tract/Block Group Substitution file. If a household does not have at least one activity location with a matching tract/block group in either the network activity location table, the Tract/Block Group Substitution file is omitted from the located population produced by *BlockGroupLoc*.

Usage:

```
BlockGroupLoc <configuration file>
```

Table 15 lists the configuration file keys from the TRANSIMS configuration file used by *BlockGroupLoc*. Some keys have default values that may be used if the key is not specified in the configuration file.

Table 15. BlockGroupLoc configuration file keys.

Configuration File Key	Description
ACT_BLOCKGROUP_HEADER	The user data column header in the network activity location file used to specify the block group. Default = BG
ACT_HOME_HEADER	The user data column header in the network activity location file used to specify single-family home locations. Default = HOME
ACT_MULTI_FAMILY_HEADER	The user data column header in the network activity location file used to specify multifamily-home locations. If not specified, multifamily user data from the activity location file is ignored.
ACT_TRACT_HEADER	The user data column header in the network activity location file used to specify the census tract. Default = TRACT
NET_ACTIVITY_LOCATION_TABLE*	The name of the network activity location table.
NET_DIRECTORY*	The directory in which the network files reside.
NET_LINK_TABLE*	The name of the network link table.
NET_NODE_TABLE*	The name of the network node table.
POP_BASELINE_FILE*	The name of the file containing the baseline population.
POP_LOCATED_FILE*	The name of the file in which the located population will be written.
POP_NEAREST_BG_FILE	The name of the Tract/Block Group Substitution file that contains information about the nearest tract/block group for block groups that have no activity locations on the transportation network.
POP_STARTING_HH_ID	The number from which the generated households will be sequentially numbered. Default = 1

Configuration File Key	Description
`POP_STARTING_PERSON_ID`	The number from which the generated persons will be sequentially numbered. Default = 101

* Configuration file keys that are required for *BlockGroupLoc*. All others are optional and will use default values.

☛ The current version of *BlockGroupLoc* must be executed once for each of the three types of synthetic household files produced by running the Population Synthesizer. Specify distinct values for `POP_BASELINE_FILE` and `POP_LOCATED_FILE` in each run. In addition, specify different values for `POP_STARTING_HH_ID` and `POP_STARTING_PERSON_ID` in each run so that IDs in the output files do not overlap.

It may be necessary to examine the successive output files to determine the last value of each ID used thus far. To do this, specify the values `Sfr-area` for `ACT_HOME_HEADER`, `Mfr-area` for `ACT_MULTI_FAMILY_HEADER`, and `BLOCKGR` for `ACT_BLOCKGROUP_HEADER`.

☛ Once *BlockGroupLoc* produces the three output files, they must be concatenated into a single file. To do this, use a text editor to remove the two header lines from the second and third files, then append each file to the first file. The resulting concatenated file will be used as an input file for the *Vehgen* program (which is described in the following section) and for the Activity Generator. There will be fewer records in the located population than were in the household files because of the network's limited size.

3.1.5 Generating a TRANSIMS Vehicle File for Located Synthetic Populations

3.1.5.1 Vehgen

A TRANSIMS vehicle file contains information about the initial locations of a household's vehicles. For most households, the vehicle's starting location will be the parking location near the household's home location.

Vehgen creates a TRANSIMS vehicle file that contains information about the household's vehicles and their starting locations.

To find each vehicle's starting parking location, iterate through the process links connected to the home activity location. Every home activity location must have at least one parking location that is accessible via the activity location's process links.

<u>Usage</u>:

```
Vehgen <configuration file>
```

Table 16 lists the configuration file keys from the TRANSIMS configuration file used by *Vehgen*. Some keys have default values that may be used if the key is not specified in the configuration file.

Table 16. Vehgen configuration file keys.

Configuration File Key	Description
NET_ACTIVITY_LOCATION_TABLE	The name of the network activity location table.
NET_DIRECTORY	The directory in which the network files reside.
NET_LINK_TABLE	The name of the link table.
NET_NODE_TABLE	The name of the network node table.
NET_PARKING_TABLE	The name of the network parking table.
NET_PROCESS_LINK_TABLE	The name of the network process-link table.
NET_TRANSIT_STOP_TABLE	The name of the network transit-stop table.
POP_LOCATED_FILE	The name of the file containing the located population.
POP_STARTING_VEHICLE_ID	The number from which the vehicle IDs will be sequentially numbered. Default = 100
VEHICLE_FILE	The name of the TRANSIMS vehicle file that will be written.

3.2 Activity Generator

3.2.1 Usage

The name of the Trip Table Activity Generator program is *ActTripGen*. Several TRANSIMS configuration file keys are provided to control the numbering of households, persons, and vehicles.

By controlling the initial numbering, users can run the Activity Generator multiple times using different trip tables and produce consistently numbered population and vehicle files. The user also can control the random number seed.

The Activity Location Table in the TRANSIMS Network must contain a column of user data that specifies the zone number associated with each activity location.

Table 17 contains configuration file keys for the Trip Table Activity Generator.

Table 17. Trip Table Activity Generator configuration file keys.

Configuration File Key	Description
ACT_TAZ_HEADER	The column header in the network activity location file that contains the zone information. Default = TAZ

Configuration File Key	Description
ACT_TRIP_TABLE_OUTPUT*	The name of the activity file that will be output from the Trip Table Activity Generator.
ACT_TRIPTABLE_FILE*	The name of the file containing the trip table matrix.
ACT_TRIPTABLE_STARTING_HH_ID	The starting household ID for households generated from trip table matrices. Default = 1
ACT_TRIPTABLE_STARTING_PERSON_ID	The starting person ID for travelers generated from trip table matrices. Default = 1
ACT_TRIPTABLE_STARTING_VEHICLE_ID	The starting vehicle ID for vehicles generated from trip table matrices. Default = 1
ACT_TRIPTABLE_VEHICLE_FILE*	The name of the vehicle file that will be output from the Trip Table Activity Generator.
ACT_TRIPTIME_FILE*	The name of the file containing the time of day trip table data.
MODE_MAP_FILE*	The name of the TRANSIMS mode file containing mapping between mode strings and integer values. The string `wcw` must be in this file.
NET_ACTIVITY_LOCATION_TABLE*	The network activity location table. Must contain a column that has the zone number for the activity locations.
NET_DIRECTORY*	The directory where the network tables reside.
NET_LINK_TABLE*	The network link table.
NET_NODE_TABLE*	The network node table.
NET_PARKING_TABLE*	The network parking table.
NET_PROCESS_LINK_TABLE*	The network process link table.
NET_TRANSIT_STOP_TABLE*	The network transit stop table (may be an empty table).
POP_TRIPTABLE_FILE*	The name of the population file that will be output from the Trip Table Activity Generator.
VEH_VEHICLE_SUBTYPE	The subtype of the vehicle fleet will be generated. Default = 0
VEH_VEHICLE_TYPE	The type of vehicles that will be generated. Default value is assigned from a type enumeration in the TRANSIMS Network = 1 (kAuto)

* Required configuration file keys. All others are optional and will use default values.

The Trip Table Activity Generator is invoked with a single command line argument, the name of the TRANSIMS configuration file.

Example:

```
% $TRANSIMS_HOME/bin/ActTripGen <configuration file name>
```

3.2.2 Activity Generator Usage

3.2.2.1 Overview

The Activity Generator uses data files that are specified in a TRANSIMS configuration file. Some data files are specific to a network and use activity patterns derived from the Portland activity survey. Data files in the TRANSIMS distribution are for the Bignet Network only.

Found in the *TRANSIMS_HOME/data/bignet/data/actgen* directory, the data files are specified in the Activity Generator configuration file, *TRANSIMS_HOME/data/bignet/bignet.cfg*. Refer to Appendix B in Volume Three (*Modules*), Chapter Three (*Activity Generator*) for a description of the configuration file keys used by the Activity Generator.

The Activity Generator requires a synthetic population that has demographics that exactly match the demographic variables in the regression tree. The Population Converter program, *PopConverter*, will create this population from the synthetic population produced by the TRANSIMS Population Synthesizer and located on the Bignet Network using the Block Group Locator program. The Population Converter program uses a TRANSIMS configuration file to specify the input population and the output file for the converted population. The configuration file *TRANSIMS_HOME/data/bignet/bignet.cfg* contains the appropriate keys to run the population converter on the test population for the Bignet Network. The test population that is located on the BignetNetwork is in *TRANSIMS_HOME/data/bignet/output/population.located*. The Activity Generator uses a TRANSIMS vehicle file to specify the vehicle IDs that are used in a household's activities. The vehicle file is created from the located population using the Vehicle Generator program. The vehicle file is *TRANSIMS_HOME/data/bignet/output/vehicles*.

3.2.2.2 How to Run

The environment variable `TRANSIMS_HOME` must be set to the directory where the TRANSIMS distribution is installed. The instructions below assume that a TRANSIMS population has been created using the TRANSIMS Population Synthesizer and located on the Bignet Network and that a TRANSIMS vehicle file has been generated from the located population.

- Change directory to *TRANSIMS_HOME/data/bignet*
  ```
  % cd $TRANSIMS_HOME/data/bignet
  ```

- Run the Population Converter.
  ```
  % $TRANSIMS_HOME/bin/PopConverter $TRANSIMS_HOME/data/bignet/bignet.cfg
  ```

- Run the Activity Generator and redirect the messages into the logfile.
  ```
  % $TRANSIMS_HOME/bin/ActivityGenerator $TRANSIMS_HOME/data/bignet/bignet.cfg  > output/ag.log
  ```

Activities for the population are produced in the *$TRANSIMS_HOME/data/bignet/output* directory. Messages from the Activity Generator will be in the file *$TRANSIMS_HOME/data/bignet/output/ag.log*.

3.3 Route Planner

3.3.1 Before Running the Route Planner

In order to run the Route Planner, ensure that all of the files specified in the configuration file keys exist (with the exception of the created output files, which are listed below). A description of all Route Planner configuration file keys is given Table 18.

Table 18. Route Planner configuration file keys.

Configuration File Key	Description
ACTIVITY_FILE	Path to a TRANSIMS activity file. **Required**.
LOG_ROUTING	Turn on Route Planner logging. This produces information about the status and progress of the Route Planner. Default = 0
LOG_ROUTING_DETAIL	Turn on detailed Route Planner logging. Produces many messages. Default = 0
MODE_MAP_FILE	Path to a mode file. **Required**.
PLAN_FILE	Name of the file where plans should be written. (Overwrites an existing file.) **Required**.
ROUTER_BIKING_SPEED	Speed to use when computing delays for walk links traversed by bicycle (meters/second). Default = 4.0
ROUTER_CORR	Floating-point number, between 0 and 1. The Route Planner will change the reported length of a link to be equal to its Euclidean length whenever the ratio of the two is less than this value. This is done in order to avoid problems when the Sedgewick-Vitter heuristic is used. Default = 0.0
ROUTER_DELAY_NOISE	Percentage of noise to add to link delays. Default = 0
ROUTER_FILTER_EXCLUDE_MODE	Plan modes not include in plan file. Default it to include no modes. Only one of INCLUDE_MODE and EXCLUDE_MODE may be specified.
ROUTER_FILTER_EXCLUDE_VEHICLE	Plan vehicle types not to include in plan file. Default is to include no vehicle types. Only one of INCLUDE_VEHICLE and EXCLUDE_VEHICLE can be specified.
ROUTER_FILTER_INCLUDE_MODE	Plan modes to include in plan file. Default is to include all modes.
ROUTER_FILTER_INCLUDE_VEHICLE	Plan vehicle types to include in plan file. Default is to include all vehicle types.
ROUTER_GET_OFF_TRANSIT_DELAY	Delay encountered when exiting a transit vehicle. Default = 4 seconds
ROUTER_GET_ON_TRANSIT_DELAY	Delay encountered when boarding a transit vehicle. Default = 3 seconds

Configuration File Key	Description
ROUTER_HOUSEHOLD_FILE	Path to a file containing a list of integer IDs for householders to be planned.
ROUTER_INTERNAL_PLAN_SIZE	Positive integer. Should be enough to accommodate the length (in number of nodes) of the shortest path between any two nodes in the network (and may need to be quite large when multimodal plans are used). Default = 400
ROUTER_LINK_DELAY_FILE	Feedback file from which to read link delays. If the key is not present or the file does not exist, the free speed delays are used.
ROUTER_MESSAGE_LEVEL	Level of warning messages to produce: -2 (ERROR) -1 (PRINT) 0 (SEVERE WARNING) 1 (WARNING). Produces information about possible anomalies the Route Planner has encountered. Default = 1
ROUTER_NUMBER_THREADS	Positive integer. Number of worker threads to be used. A value of 0 means no threads will be used. Default = 0
ROUTER_OVERDO	Non-negative floating-point number. If set to 0, no adjustment is made to the distance estimates. If positive, the search for the shortest path to the origin will be biased in the direction of a straight line to the destination. This will produce non-optimal paths. The paths will still be reasonable, but the heuristic may cause relatively small congestion on links to be ignored, and this can break the iterative relaxation mechanism. Default = 0.0
ROUTER_PROBLEM_FILE	Path name to a file in which activities with anomalies identified by the Route Planner are written. **Required.**
ROUTER_SEED	Seed to use for random number generator. If key is set to 0, use process ID. Default = 0
ROUTER_WALKING_SPEED	Speed to use when computing delays for walk links (meters/second). Default = 1.0
ROUTER_ZERO_BACKD	Integer, 0 or 1. Default = 0
TRANSIT_ROUTE_FILE	File containing route of transit vehicles.
TRANSIT_SCHEDULE_FILE	File containing schedules of transit vehicles.
VEHICLE_FILE	Path to a TRANSIMS vehicle file. **Required**.

3.3.2 Required Input Files

The required input files for the Route Planner are listed in Table 19.

Table 19. Required input files for the Route Planner.

Input File	Description
ROUTER_ACTIVITY_FILE	Contains the activities for the travelers to be routed.
MODE_MAP_FILE	Contains the travel modes to be used for routing.
VEHICLE FILE	Contains the starting locations of available vehicles.

If the transportation network contains transit, TRANSIT_ROUTE_FILE and TRANSIT_SCHEDULE_FILE are also required.

3.3.3 Created Output Files

The created output files for the Route Planner are listed in Table 20.

Table 20. Created output files.

Input File	Description
PLAN_FILE	Contains the plans for all successfully routed travelers.
ROUTER_PROBLEM_FILE	Contains information about unsuccessfully routed travelers.

3.3.4 Setting the Number of CPUs

On a multiprocessor machine, the configuration file key ROUTER_NUMBER_THREADS should be set to the number of CPUs available.

On a single processor machine, ROUTER_NUMBER_THREADS should be set to 0.

3.3.5 Running the Route Planner

Run the Route Planner by keying

```
% <Router> <config>
```

where <Router> is the name of the Route Planner executable (usually Router), and <config> is the name of the configuration file to be used.

4. TRAFFIC MICROSIMULATOR

4.1.1 Parallel Configuration and Communication Setup

Running the Traffic Microsimulator requires that the user's environment be properly set up. Because it uses the Parallel Virtual Machine (PVM) or Message Passing Interface (MPI) protocols to run in a distributed environment, the user must ensure that these are correctly configured before using the Microsimulator. This section gives some tips on setting up the environment.

The Microsimulator uses a master/slave paradigm to run in distributed mode. That is, there will be several copies of the executable running during a simulation run. One of these will be the "master," responsible for providing a synchronization heartbeat to the slaves and for other chores such as collecting, sorting, and writing output. These notes are written in the expectation that the master and slaves will all be running on the same "machine". In other words, they will be distributed over the processors of a multi-processor computer. It is also possible to run the simulator distributed over several different machines on a network.

Communication among the master and slaves is accomplished by passing messages using one of two message passing protocols: MPI or PVM. Both are widely available with no license fee. Traffic Microsimulator performance is similar with either. Each communication scheme requires the user to specify the architecture used for communication, and MPI further requires the user to choose the "device" to be used for communication.

For example, on Solaris under PVM, an appropriate architecture is SUN4SOL2; for MPI under Linux, an appropriate architecture is LINUX and device is ch_p4. See the documentation supplied with PVM and MPI for further description of allowed architectures and devices.

The choice of architecture and device is made by the user at compile time, by supplying the `PVM_ARCH` or `MPI_ARCH` and `MPI_DEVICE` command line arguments to *make*. The Makefile system creates different versions of the "CA" executable that are placed in communication environment, architecture, and device-dependent directories under the *bin* directory.

For example, compiling with the command

```
make APP=CA COMMUNICATION=PVM PVM_ARCH=SUN4SOL2
```

creates the executable *bin/ARCH.PVM.SUN4SOL2/CA.*

Similarly,

`make APP=CA COMMUNICATION=MPI MPI_ARCH=LINUX MPI_DEVICE=ch_p4`
creates *bin/ARCH.MPI.LINUX.ch_p4/CA*. See the installation manual for more details.

Both PVM and MPI require that the user gain access to the machine on which processes are to be run without typing a password. The exact mechanism for spawning processes depends on the implementation and installation of these packages. One scheme which should work for all installations uses a file named *.rhosts* in the user's home directory. This file should list the name of the machine to be used and the user's login name on that machine.

The *.rhosts* file's permissions should be set to deny access to others. You should be able to remote shell (*rsh*) to the desired machine without being prompted for a password. Because of potential security risks, some networks automatically remove *.rhosts* files on a daily basis, so it may be necessary to recreate it. Contact your system administrator for details on the most appropriate setup for your system.

How the CA is invoked depends on the communication protocol in use.

4.1.1.1 PVM

The PVM daemon must be running before the CA is invoked. The daemon can be started by typing `pvm`. At the PVM prompt, the user can type

- `ps` to see a list of all the user's processes currently running under PVM;
- `quit`, to exit the console, leaving the daemon running; or
- `halt` to kill the daemon and all the user's processes currently running under PVM.

PVM must be restarted each time the machine is rebooted.

It is important to make sure that the version of PVM running is the same as the one the CA was linked with. It is recommended to set the environment variable `PVM_ROOT` to *$TRANSIMS_HOME/pvm/pvm3*. In addition, make sure *$PVM_ROOT/bin* and *$PVM_ROOT/lib* are both in your path.

The PVM version of the executable requires that the configuration file key `PAR_HOST_0` be defined in the configuration file. It should be set to the name of the machine on which the executable is invoked.

When PVM is running, invoke the simulator using the full pathname to the executable and to a configuration file on the command line. Assuming the configuration file is named *test.cfg* in the current working directory, and that the communication architecture is SUN4SOL2, the simulator could be started with the following command line:

```
$TRANSIMS_HOME/bin/ARCH.PVM.SUN4SOL2/CA  test.cfg
```

An optional additional argument is the name of a file the CA will use for its logging output in the directory specified by the configuration file key `OUT_DIRECTORY`. The default value is *logfile*. To send logging information to the file *Microsim.log*, the example above command line would become:

```
     $TRANSIMS_HOME/bin/ARCH.PVM.SUN4SOL2/CA  test.cfg
Microsim.log
```

4.1.1.2 MPI

MPI jobs are usually started with a command called `mpirun`, which determines information about your machine configuration and starts all of the processes. This script is distributed with MPI and can vary from one implementation to another. It usually requires the argument `-np <integer>`, which specifies the number of processes to spawn. This should be set to one more than the number of slaves specified by the configuration file key `PAR_SLAVES`. The other arguments to `mpirun` are the full pathname of the executable and the arguments for the executable.

Compiling the MPI executable requires the environment variable `MPI_ROOT` to be set correctly. If you wish to use the MPI distributed with TRANSIMS, set `MPI_ROOT` to *$TRANSIMS_HOME/mpich-1.2.0*. Make sure that *$MPI_ROOT/bin* is in your path.

Currently, the MPI implementation requires that the full path name to the configuration file be passed in on the command line. Assuming that the configuration file is named *test.cfg* in the current working directory, that you are running with only one slave, and that you are using the communication architecture LINUX and device ch_p4, the simulator could be started with the following command line:

```
     mpirun -np 2 $TRANSIMS_HOME/bin/ARCH.MPI.LINUX.ch_p4/CA
`pwd`/test.cfg
```

As above, you may specify the name of a log file in the `OUT_DIRECTORY` by adding one more argument to this command line.

The message "Permission denied" usually means that `mpirun` cannot start up slaves. Often this happens because there is no *.rhosts* file. This file should be in your home directory, should contain the name of the machine you will run on and your user name, and should be accessible only to you.

4.1.2 Msim.csh Script

In addition to invoking the executable directly, it may be invoked using the script *Msim.csh* located in *$TRANSIMS_HOME/scripts*. Because it tries to handle many different configurations and user environments, using the script is not as robust as invoking the executable directly. Nonetheless, it should work in the majority of cases as long as the following conditions are met:

- `csh` is in */bin*.
- `awk` is in your path.
- `SetEnv` is in *$TRANSIMS_HOME/bin*.
- Your *cshrc* file does not contain any `echo` commands.

- The environment variable TRANSIMS_HOME is set.

In addition, the environment variable PAR_COMMUNICATION should be set to either PVM or MPI, and the environment variable PAR_DEVICE should be set to an appropriate architecture or architecture/device combination for MPI, such as LINUX.ch_p4. If they are not set, the default values MPI and solaris.ch_p4 will be used. If PVM is used, the script tries to ensure that PAR_HOST_0 is set correctly.

The *Msim.csh* script takes one optional argument—the name of a configuration file to use. If this is not provided, the script attempts to derive it from the name of the current working directory. For example, if you run *Msim.csh* from the directory *$TRANSIMS_HOME/data/bignet*, it will assume the configuration file is *$TRANSIMS_HOME/data/bignet/bignet.cfg*. The script runs the Traffic Microsimulator with the log file name set to *Microsim.log*.

4.1.3 Configuration Parameters

Configuration parameters control how drivers and vehicles behave in traffic. Variations in behavior among drivers are accomplished by allowing certain behaviors to vary randomly within limits. Table 21 lists and describes configuration parameters

Table 21. Configuration parameters.

Configuration File Key	Description
CA_BROADCAST_ACC_CPN_MAP CA_BROADCAST_TRAVELERS	If Broadcast Travelers is set, migrating travelers are broadcast to every CPU. Because only one CPU will eventually make use of the traveler, this is inefficient. If Broadcast Acc CPN Map is set, each CPU knows which CPU is associated with every accessory, so traveler migration messages can be targeted to only the single CPU that needs them. If the CPN Map is not broadcast, travelers must be broadcast.
CA_DECELERATION_PROBABILITY	To enhance traffic variation, each automobile driver randomly decides whether to decelerate for no apparent reason at each timestep. The probability of decelerating is a value in the range 0.0 to 1.0. Default = 0.2
CA_ENTER_TRANSIT_DELAY CA_EXIT_TRANSIT_DELAY	These keys specify the mean number of timesteps it takes for a single traveler to enter or exit a transit vehicle.

Configuration File Key	Description
`CA_GAP_VELOCITY_FACTOR`	At unsignalized intersections and during protected movements at signalized intersections, drivers wait for a suitable gap in cross traffic before proceeding through the intersection. The number of empty cells in a suitable gap is based on the speed of the cross traffic and the gap velocity factor. The suitable gap is calculated for each lane of the cross traffic. `Gap = Speed of Oncoming Vehicle * Gap Velocity Factor` The gap velocity factor must be greater than 0.0. The default value is 3.0. Note that vehicles with a speed of 0 result in a suitable gap size of 0, which improves traffic flow in congested conditions.
`CA_IGNORE_GAP_PROBABILITY`	Drivers at unsignalized intersections wait for a suitable gap in cross traffic before proceeding through the intersection. Allowing each driver to ignore the gap constraint with some probability prevents the deadlock that would take place when vehicles are waiting for each other at multiway stop/yield signs. The probability that the drivers at multiway stop/yield signs will ignore the constraint is a value in the range of 0.0 to 1.0. Default = 0.66
`CA_INTERSECTION_CAPACITY`	`Intersection Capacity` determines the number of vehicles that can be held by each intersection's buffers.
`CA_INTERSECTION_WAIT_TIME`	`Intersection Wait Time` specifies the number of seconds that a vehicle requires to pass through a signalized intersection. A vehicle resides in an intersection-queued buffer for this amount of time and is then placed on the next link if the first cell on that link is unoccupied. It will remain in the intersection for a longer time if entry to the next link is blocked by another vehicle. Valid values are positive. Default = 1 second
`CA_LANE_CHANGE_PROBABILITY`	Variation in traffic is reduced by not allowing every driver who would change lanes based on vehicle speed and gaps in the traffic to do so at each timestep. This is done to prevent *lane hopping*. The probability that a driver will change lanes when speed and gaps permit is a value in the range of 0.0 to 1.0. Default = 0.99
`CA_LATE_BOUNDARY_RECEPTION`	If `Late Boundary` Reception is set, the simulation will try to overlap computation and communication.

Configuration File Key	Description
`CA_LOOK_AHEAD_CELLS`	The preferred lane for a vehicle to be in as it approaches an intersection depends on the connectivity from the current link to the next link in the plan. In some situations, it is advantageous for the driver to look beyond the next link to subsequent links in the plan when deciding the preferred lane. Look Ahead Cells controls how far ahead the driver will look. A value of 0 indicates that the driver will not look beyond the next link. A positive value indicates that the driver will look at least one additional step beyond the next step in the plan. The number of additional links considered is determined by the lengths of the subsequent links, with link lengths being summed until the accumulated distance is greater than or equal to Look Ahead Cells. Valid values are positive or zero. Default = 35 cells
`CA_MAX_WAITING_SECONDS`	`Max Waiting Seconds` determines the number of seconds that a vehicle will try to enter an intersection. If the vehicle has not moved from the link into or through the intersection in `Max Waiting Seconds`, the vehicle abandons its plan and tries an alternative movement through the intersection (if one exists). `Max Waiting Seconds` must be > 0 and should be greater than the longest red phase of the traffic controls in the simulation. Default = 600 seconds
`CA_NO_TRANSIT`	If this flag is set, travelers whose plans originate or end at a transit stop are removed from the simulation. None of their remaining legs are used. (The transit driver plans do not fall into this category, thus transit vehicles can still be present in the simulation, but no passengers will use them.)
`CA_OFF_PLAN_EXIT_TIME`	`Off Plan Exit Time` specifies the number of seconds a vehicle is allowed to deviate from its plan before being removed from the simulation. This prevents off-plan vehicles from wandering on the transportation network. Valid values are positive. Default = 1 second
`CA_PLAN_FOLLOWING_CELLS`	`Plan Following Cells` specifies a count of the number of cells preceding the intersection within which a vehicle will make lane changes to get in an appropriate lane and thus transition to the next link in its plan. Beyond this distance, lane-changing decisions are based only on vehicle speed and gaps in the traffic. Within this distance, the lane required by the vehicle's plan is also taken into account. As the vehicle nears the intersection, the bias to be in the lane required to stay on plan is increased. Valid values are positive or zero. Default = 70 cells

Configuration File Key	Description
CA_RANDOM_SEED1 CA_RANDOM_SEED2 CA_RANDOM_SEED3	These three values are combined to initialize the random number generator. Note that the actual sequence of random numbers generated on a slave also depends on the number of slaves and the partitioning in general.
CA_SEQUENCE_LENGTH	The slaves are implicitly synchronized among themselves by the actions of passing boundaries and migrating vehicles. They are also explicitly synchronized by the master every Sequence Length timestep. It may be more efficient to allow the implicit synchronization to control the simulation.
CA_SIM_START_HOUR CA_SIM_START_MINUTE CA_SIM_START_SECOND	These values are combined to calculate the simulation's starting time. Plans whose estimated arrival time is before the start time are not executed.
CA_SIM_STEPS	The simulation executes Sim Steps timesteps before exiting.
CA_SLAVE_MESSAGE_LEVEL CA_MASTER_MESSAGE_LEVEL	Only warning messages whose severity is at least as high as `Message Level` will be written to the master or slave log file.
CA_SLAVE_PRINT_MASK CA_MASTER_PRINT_MASK	These variables control which logging messages to ignore. They are code set within the code based on the values of the `LOG_` configuration file keys and should not be set directly.
CA_TRANSIT_INITIAL_WAIT	`Transit Initial Wait` specifies the number of timesteps a transit vehicle must be present at a transit stop before any passengers get on or off.
CA_USE_NETWORK_CACHE	If set, use a cached binary representation of the network. This representation would have been created by a prior run of the simulation.
CA_USE_PARTITIONED_ROUTE_ FILES	It is more efficient for slaves to read only those plans that start in the part of the network for which they are responsible. If the partitioning to be used by the simulation is available (for example, from a prior run of the simulation), the *DistributePlans* utility will create a separate pair of indexes for each slave into one common plan file. If `Use Partitioned Route Files` is set, the slaves will look for these slave-specific indexes. If they do not exist, the simulation will fall back to using a single global pair of indexes.
CA_USE_ROMIO_FOR_OUTPUT	If `Use Romio For Output` is set, and the executable was compiled with the `USE_ROMIO` and `USE_MPI` flags defined, the parallel output system will use ROMIO files instead of Unix files.
PAR_HOST_COUNT	The number of distinct machines that make up the parallel machine environment.

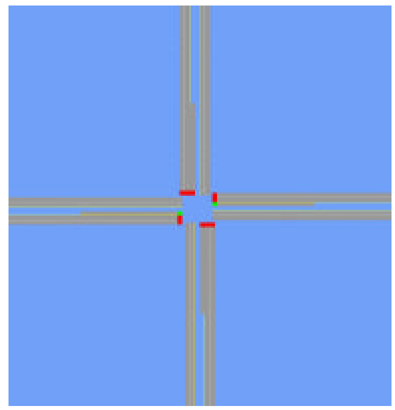
Configuration File Key	Description
PAR_HOST_I PAR_HOST_CPUS_I PAR_HOST_SPEED_I	These variables describe the parallel machine environment to the simulation. There should be one set of these three variables, with I replaced by an integer from 0 to the value of PAR_HOST_COUNT – 1, for each host. Host should be a string containing the name of the machine. Host CPUs should give the number of CPUs available for use on the machine. Host Speed should give the relative speeds of the different machines in arbitrary units. The sum of all the values of Host CPUs must be at least one larger than the number of slaves requested.
PAR_RTM_INPUT_FILE RTM_FEEDBACK_FILE RTM_SAMPLE_INTERVAL PAR_RTM_PENALTY_FACTOR	The partitioning algorithms try to find the partition that spreads the computation associated with nodes and links evenly while simultaneously trying to minimize the communication costs associated with split links. The costs for each node and link can be estimated using run time costs from prior runs. These costs are sampled at the interval defined by RTM Sampling Interval and written out to the file named by RTM File. They are read in from the file found in the directory named by OUTPUT_DIRECTORY.
PAR_SLAVES	This key sets the number of slave processes to spawn. It must be smaller than the number of host CPUs available (to allow one process for the master).
PLAN_FILE	The plan file specifies the name of the file in which plans reside or a string to which *.tim.idx* and *.trv.idx* can be appended to find the time-sorted and traveler-id-sorted indexes into a plan file(s). The plans should include all travelers; for example, plans created by the Route Planner, transit driver plans, freight plans, etc. The name should be given as an absolute path name because the slave executables are not always run from the current working directory.
VEHICLE_FILE	The vehicle file specifies the name in which vehicles reside or a string to which *.veh.idx* can be appended to find the vehicle-id-sorted index into a vehicle file(s). The vehicle file must include all vehicles to be used in the simulation.
VEHICLE_PROTOTYPE_FILE	The vehicle prototype file must include information about every vehicle type used in the simulation.

4.2 Generic Signal Tutorial

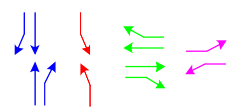
This tutorial outlines how to create generic signals for calibration and testing, and how to run the TRANSIMS Traffic Microsimulator to measure their performance. The files for the tutorial reside in the directory *$TRANSIMS_HOME/data/gensig*.

4.2.1 Generic Signal Builder

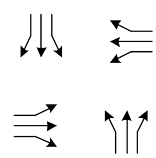
The generic signal builder is a software application for constructing an intersection and vehicular demand for testing the performance of an actuated signal in the TRANSIMS microsimulator. It generates the following:



- One intersection with an actuated signal and a specified timing plan.
- Four links in the cardinal directions with a specified length, number of lanes, and speed limit in the north-south and east-west directions.
- Four pocket lanes (one per link) of a specified length for left turns.
- Eight vehicle detectors (two per phase) of a specified length and placement.
- Four phases at the signal: “north-south through,” “north-south left,” “east-west through,” and “east-west left.”



- Plans for vehicles with specified headways for the twelve possible movements at the intersection: “southbound left,” “southbound through,” “southbound right,” “westbound left,” “westbound through,” “westbound right,” “northbound left,” “northbound through,” “northbound right,” “eastbound left,” “eastbound through,” and “eastbound right.”



The TRANSIMS configuration file keys controlling the details of the intersection and demand are given in Table 22. The values in the table reflect those used in this tutorial.

Table 22. Generic Signal Builder configuration file keys used in example.

Configuration File Key	Value in Example	Description
CALIB_PLUS_DETECTOR_LENGTH	37.5	The length [meters] of the detectors. Default = six cells
CALIB_PLUS_DETECTOR_OFFSET	<default >	The offset [meters] of the detectors, measured from the point of the detector closest to the node to the node. Default = the intersection setback

Configuration File Key	**Value in Example**	**Description**
CALIB_PLUS_GREEN_EXTENSION	0.60	The green extension in terms of the fraction of the initial green. Default = 0.50
CALIB_PLUS_GREEN_LEFT_EW	8	The number of seconds for greens of the “east-west left” phase. The phase does not exist if this is zero.
CALIB_PLUS_GREEN_LEFT_NS	8	The number of seconds for greens of the “north-south left” phase. The phase does not exist if this is zero.
CALIB_PLUS_GREEN_THRU_EW	20	The number of seconds for greens of the “east-west through” phase. The phase does not exist if this is zero.
CALIB_PLUS_GREEN_THRU_NS	20	The number of seconds for greens of the “north-south through” phase. The phase does not exist if this is zero.
CALIB_PLUS_LANES_EW	2	The number of lanes in the east-west direction.
CALIB_PLUS_LANES_NS	2	The number of lanes in the north-south direction.
CALIB_PLUS_LINK_LENGTH	<default >	The length [meters] of the links. Default = thirty cells plus the intersection setback
CALIB_PLUS_PARKING_OFFSET	<default >	The offset [meters] of the parking from the edges of the network.
CALIB_PLUS_POCKET_LENGTH	<default >	The length [meters] of pocket lanes. Default = eight cells
CALIB_PLUS_SETBACK	<default >	The setback [meters] at the intersection. This defaults to an optimal value for the Output Visualizer.
CALIB_PLUS_SPACING_EAST_LEFT	18	The spacing [seconds] for vehicles leaving parking and planing to make a left turn movement approaching the intersection from the east.
CALIB_PLUS_SPACING_EAST_RIGHT	18	The spacing [seconds] for vehicles leaving parking and planing to make a right turn movement approaching the intersection from the east.
CALIB_PLUS_SPACING_EAST_THRU	4	The spacing [seconds] for vehicles leaving parking and planing to make a through movement approaching the intersection from the east.
CALIB_PLUS_SPACING_NORTH_LEFT	0	The spacing [seconds] for vehicles leaving parking and planing to make a left turn movement approaching the intersection from the north.

Configuration File Key	Value in Example	Description
CALIB_PLUS_SPACING_NORTH_RIGHT	0	The spacing [seconds] for vehicles leaving parking and planing to make a right turn movement approaching the intersection from the north.
CALIB_PLUS_SPACING_NORTH_THRU	3	The spacing [seconds] for vehicles leaving parking and planing to make a through movement approaching the intersection from the north.
CALIB_PLUS_SPACING_SOUTH_LEFT	24	The spacing [seconds] for vehicles leaving parking and planing to make a left turn movement approaching the intersection from the south.
CALIB_PLUS_SPACING_SOUTH_RIGHT	21	The spacing [seconds] for vehicles leaving parking and planing to make a right turn movement approaching the intersection from the south.
CALIB_PLUS_SPACING_SOUTH_THRU	7	The spacing [seconds] for vehicles leaving parking and planing to make a through movement approaching the intersection from the south.
CALIB_PLUS_SPACING_WEST_LEFT	30	The spacing [seconds] for vehicles leaving parking and planing to make a left turn movement approaching the intersection from the west.
CALIB_PLUS_SPACING_WEST_RIGHT	27	The spacing [seconds] for vehicles leaving parking and planing to make a right turn movement approaching the intersection from the west.
CALIB_PLUS_SPACING_WEST_THRU	24	The spacing [seconds] for vehicles leaving parking and planing to make a through movement approaching the intersection from the west.
CALIB_PLUS_SPEED_EW	<default>	The speed limit [meters per second] in the east-west direction. Default = five cells per second
CALIB_PLUS_SPEED_NS	<default>	The speed limit [meters per second] in the north-south direction. Default = five cells per second.
CALIB_PLUS_TIME_LIMIT	3600	The number of seconds for which to generate plans.

The generic signal builder also uses the additional TRANSIMS keys in Table 23—not all of these are required, however.

Table 23. Additional Generic Signal Builder configuration file keys.

Configuration File Key	Required?
CA_CELL_LENGTH	no
NET_ACTIVITY_LOCATION_TABLE	yes
NET_BARRIER_TABLE	yes
NET_DETECTOR_TABLE	yes
NET_DIRECTORY	yes
NET_LANE_CONNECTIVITY_TABLE	yes
NET_LANE_USE_TABLE	yes
NET_LANE_WIDTH	no
NET_LINK_MEDIAN_HALFWIDTH	no
NET_LINK_TABLE	yes
NET_NODE_TABLE	yes
NET_PARKING_TABLE	yes
NET_PHASING_PLAN_TABLE	yes
NET_POCKET_LANE_TABLE	yes
NET_PROCESS_LINK_TABLE	yes
NET_SIGNAL_COORDINATOR_TABLE	yes
NET_SIGNALIZED_NODE_TABLE	yes
NET_SPEED_TABLE	yes
NET_STUDY_AREA_LINKS_TABLE	yes
NET_TIMING_PLAN_TABLE	yes
NET_TRANSIT_STOP_TABLE	yes
NET_TURN_PROHIBITION_TABLE	yes
NET_UNSIGNALIZED_NODE_TABLE	yes
OUT_SNAPSHOT_LINKS_1	yes
OUT_SNAPSHOT_NODES_1	yes
PLAN_FILE	yes
VEHICLE_FILE	yes
VEHICLE_PROTOTYPE_FILE	yes

The generic signal builder is invoked with a single argument:

```
BuildTestSignal <config-file>
```

where `<config-file>` is the name of the configuration file containing the keys listed in the two tables above. The directory *$TRANSIMS_HOME/data/gensig* contains the configuration file *signal.cfg* with the keys used in this tutorial. Running *BuildTestSignal* with this configuration file results in the network files stored in the *network* subdirectory and in the vehicle and plan files stored in the *data* subdirectory.

4.2.2 Assessing Generic Signal Response

In order to assess the response of the generic signal created by the builder, we run the Traffic Microsimulator and analyze the output. The script *signal.csh* in the directory *$TRANSIMS_HOME/data/gensig/scripts* runs the does this. After running the Traffic Microsimulator, the *output* subdirectory contains several files of interest:

- *snapshot.veh* – the vehicle snapshot data for the travelers—suitable for viewing in the Output Visualizer
- *snapshot.sig* -- the signal snapshot data for the actuated signal—suitable for viewing in the Output Visualizer
- *times.trv* – the travel time and inconvenience measure statistics for the travelers—suitable for analysis with the *scripts/counts.awk* script
- *lost.trv* – traveler event data for any travelers that became lost in the simulation—there should be few or none

The script will also produce console output summarizing the response of the signal. Four performance measures are tabulated for each movement at the intersection:

- *flow [veh/hr]* – the flow of vehicles making the movement
- *travel time [s/veh]* – the observed mean and standard deviation for the travel times measured from when a vehicle leaves its parking location on the incoming link to when it enters its parking location on the outgoing link
- *time stopped [s/veh]* – the observed mean and standard deviation for the number of seconds that a vehicle is stopped while it waits to pass through the intersection
- *accelerations from stop [#/veh]* – the observed mean and standard deviation for the number of times a vehicle has to accelerate after being stopped on the roadway

For the example in this tutorial, the summary should look similar to the following:

```
Movement:                    southbound, left
  Flow [veh/hr]:             0

Movement:                    southbound, thru
  Flow [veh/hr]:             1177
  Travel Time [s/veh]:       35.3696 (mean), 15.3534 (s.d.)
  Time stopped [s/veh]:      3.19031 (mean), 9.24733 (s.d.)
  Accels. from stop [#/veh]: 2.13594 (mean), 0.859248 (s.d.)

Movement:                    southbound, right
  Flow [veh/hr]:             0

Movement:                    westbound, left
  Flow [veh/hr]:             197
  Travel Time [s/veh]:       86.4162 (mean), 43.3299 (s.d.)
```

```
  Time stopped [s/veh]:      11.1015 (mean), 25.0976 (s.d.)
  Accels. from stop [#/veh]: 3.30457 (mean), 1.48758 (s.d.)

Movement:                    westbound, thru
  Flow [veh/hr]:             894
  Travel Time [s/veh]:       52.4586 (mean), 26.8092 (s.d.)
  Time stopped [s/veh]:      2.84228 (mean), 9.19959 (s.d.)
  Accels. from stop [#/veh]: 2.9396 (mean), 1.5013 (s.d.)

Movement:                    westbound, right
  Flow [veh/hr]:             199
  Travel Time [s/veh]:       54.5578 (mean), 28.1368 (s.d.)
  Time stopped [s/veh]:      5.09045 (mean), 13.0257 (s.d.)
  Accels. from stop [#/veh]: 2.59799 (mean), 1.32543 (s.d.)

Movement:                    northbound, left
  Flow [veh/hr]:             139
  Travel Time [s/veh]:       150.367 (mean), 87.6663 (s.d.)
  Time stopped [s/veh]:      17.3813 (mean), 43.9112 (s.d.)
  Accels. from stop [#/veh]: 3.54676 (mean), 2.14078 (s.d.)

Movement:                    northbound, thru
  Flow [veh/hr]:             499
  Travel Time [s/veh]:       67.6212 (mean), 55.6657 (s.d.)
  Time stopped [s/veh]:      4.11222 (mean), 10.2163 (s.d.)
  Accels. from stop [#/veh]: 3.11824 (mean), 2.12748 (s.d.)

Movement:                    northbound, right
  Flow [veh/hr]:             167
  Travel Time [s/veh]:       66.9701 (mean), 57.4042 (s.d.)
  Time stopped [s/veh]:      3.07784 (mean), 8.74332 (s.d.)
  Accels. from stop [#/veh]: 2.8024 (mean), 1.66552 (s.d.)

Movement:                    eastbound, left
  Flow [veh/hr]:             119
  Travel Time [s/veh]:       61.8739 (mean), 36.7648 (s.d.)
  Time stopped [s/veh]:      18.1008 (mean), 30.0298 (s.d.)
  Accels. from stop [#/veh]: 1.72269 (mean), 0.649959 (s.d.)

Movement:                    eastbound, thru
  Flow [veh/hr]:             149
  Travel Time [s/veh]:       33.2886 (mean), 19.4514 (s.d.)
  Time stopped [s/veh]:      10.0067 (mean), 15.5748 (s.d.)
  Accels. from stop [#/veh]: 1.39597 (mean), 0.555304 (s.d.)

Movement:                    eastbound, right
  Flow [veh/hr]:             133
  Travel Time [s/veh]:       33.0752 (mean), 19.1769 (s.d.)
  Time stopped [s/veh]:      10.1955 (mean), 15.1164 (s.d.)
  Accels. from stop [#/veh]: 1.43609 (mean), 0.655436 (s.d.)
```

For reference, the demand in this example is given in Table 24.

Table 24. Movement demand.

Movement	Demand [veh/hr]
Southbound, left	0
Southbound, through	1200
Southbound, right	0
Westbound, left	200
Westbound, through	900
Westbound, right	200
Northbound, left	150
Northbound, through	514
Northbound, right	171
Eastbound, left	120
Eastbound, through	150
Eastbound, right	133

We can see that nearly all of the vehicles make it through the intersection with the actuated signal.

By editing the `CALIB_PLUS_SPACING` parameters in the configuration file and then rerunning *BuildTestSignal*, one can evaluate additional demands for the signal under consideration. One can change the actuation algorithm's parameters by editing its configuration parameters:

```
NET_DETECTOR_PRESENCE_SAMPLE_TIME
NET_ACTUATED_ALGORITHM_B_BETA
NET_ACTUATED_ALGORITHM_B_DENSITY_CONST
NET_ACTUATED_ALGORITHM_B_FLOW_CONST
```

The signal timing and detector layout may be altered with other `CALIB_PLUS` configuration file keys.

4.2.3 Troubleshooting

The following are possible problems that might arise:

- The *$TRANSIMS_HOME* environment variable has not been set.
- You tried to run the microsimulator, but do not have write permission to the output directory *$TRANSIMS_HOME/data/gensig/output*.
- You tried to run *BuildTestSignal*, but do not have write permissions for the network, plan, vehicle, or output files specified in the configuration file.
- You have edited the configuration file *$TRANSIMS_HOME/data/gensig/signal.cfg* to change some of the file names or locations, but have not made the corresponding changes in the script *$TRANSIMS_HOME/data/gensig/scripts/signal.csh*.

- You changed the specification for the travel time output in the configuration file *$TRANSIMS_HOME/data/gensig/signal.cfg*–this prevents the script
- *$TRANSIMS_HOME/data/gensig/scripts/counts.awk* from running because it depends on the details of the output specification.
- The C shell executable is not in */bin/csh* or the NAWK executable is not in */bin/nawk* on your system.
- The NAWK executable on your system is a variant of the standard GNU/Linux version.

4.3 Selectors/Iteration Databases

The Selector/Iteration Database takes a single command-line argument: the TRANSIMS configuration file.

Table 25 the configuration parameters that the Selector/Iteration Database uses: these are the various reassignment and rerouting thresholds, as well as the location of the iteration database file.

Table 25. Selectors/Iteration Databases configuration file keys.

Configuration File Key	Description
ACT_FEEDBACK_FILE	The traveler IDs and generator command output.
ACTIVITY_FILE	The (index of) activities.
OUT_DIRECTORY	The directory containing output data created by the simulation.
OUT_EVENT_NAME_1	The traveler event data created by microsimulation.
PLAN_FILE	The (index of) plans.
ROUTER_HOUSEHOLD_FILE	The household IDs that need to be rerouted (includes all households with travelers included in the ACT_FEEDBACK_FILE).
SEL_FILL_ITDB	Whether to update the iteration database.
SEL_FRUSTRATION_THRESH	f_{remode}
SEL_ITDB_FILE	The iteration database file.
SEL_REASSIGN_FRAC	$f_{reassign}$
SEL_REMODEFRAC	f_{remode}
SEL_REROUTE_FRAC	$f_{reroute}$
SEL_RETIME_FRAC	f_{retime}
SEL_USE_<field>	Directs the Selector/Iteration Database to create a corresponding field in the ITDB and fill it with data.

4.4 Emissions Estimator

Before running the Emissions Estimator, the Traffic Microsimulator must be run to collect velocity summary data.

The Emissions Estimator works on velocity summary data only.

Run *ConvertVELfile* in order to convert the microsimulation velocity summary data file into a format that can be read into the Emissions Estimator.

```
% ConvertVELfile <configFilename>
```

The configuration file keys `EMISSIONS_LDV_VELOCITY_FILE` and `EMISSIONS_HDV_VELOCITY_FILE` can be used to specify the output files. If not used, the filenames default to *velocity.ldv.out* and *velocity.hdv.out*.

The configuration file keys `EMISSIONS_MICROSIM_LDV_VELOCITYFILE` and `EMISSIONS_MICROSIM_HDV_VELOCITYFILE` can be used in the configuration file to define the velocity summary output files to be used as input. If not used, the filenames default to *summary.ldv.out* and *summary.hdv.out*. These *summary.out* filenames are created by adding *.vel* to the `OUT_SUMMARY_NAME_n` file specification, where `VELOCITY` was specified by `OUT_SUMMARY_TYPE_n`.

Run the Emissions Estimator to produce a file of emissions that can be read into the Output Visualizer. The configuration file used to run the Traffic Microsimulator to collect velocity data must be specified on the command line. Before running this program, verify that all of the necessary input files are present in the directory you will be running from or that all the Emissions Estimator configuration file keys are set properly. These include *ARRAYP.INP*, *arrayp.out*, and *vehcold.dis* files, and the *velocity.ldv.out* file produced above. Refer to Volume Three (*Modules*), Chapter Seven (*Emissions Estimator*) for detailed explanations of the various input and output files.

```
% EmissionsEstimator <configFilename>
```

The file *emissions.out* will be produced. If the configuration file key `EMISSIONS_WRITE_DEGBUG_OUTPUT` is defined and set to 1, the debugging files will be written also.

Run the Output Visualizer to display the emissions. The Output Visualizer needs the configuration file used to run the microsimulation in order to know which network to bring up.

```
% Vis <configFilename>
```

Within the Output Visualizer,

- select File→Open Variable Size Box Data
- choose the *emissions.ldv.out* file produced by the Emissions Estimator

To cycle through the different emissions columns (VTT, NOX, CO, HC, FE, and FLUX), select View→Increment Column. The type of data being display is labeled in the lower window next to the time. To animate the emissions data through different timestep, use the animation tools at the bottom of the Output Visualizer's window.

To display three-dimensional emissions where not only the color of the summary box but also the height depends on the emissions value:

- rotate the network so the Z is at about 50 and the X to about 300,
- select Modes→Lights to get better shading,
- select View→3D Plans, then
- click [OK].

Just start cycling through the emissions data fields (View→Increment Column) to view the different data columns. The three-dimensional emissions can also be animated through the timesteps.

4.4.1 Troubleshooting

4.4.1.1 Messages and ERRORs from *ConvertVELfile*

Usage: `ConvertVELfile <configFilename>`

ConvertVELfile reads in a Microsimulation Velocity Summary output file, processes the data, and then writes out a file for input into the Emissions codes. This message occurs when the configuration file is not specified when running the *ConvertVELfile* program. To run *ConvertVELfile*, you must specify the configuration file on the command line.

ERROR in [ENV]:ConvertVELfile: *Unable to open input file: <velocityFilename>*

This message occurs when the *ConvertVELfile* program is either unable to locate or unable to open either of the input files specified either in the configuration file or the default files. Make sure that you have the correct location for your input file and that it has proper read permissions set.

ERROR in [ENV]:ConvertVELfile: *Unable to open output file: outputFilename*

This message occurs when the *ConvertVELfile* program is unable to open the output files specified in the configuration file or the default files. This may occur because of lack of disk space or lack of write privileges in the directory in which you are attempting to run *ConvertVELfile* or the directory specified by the configuration file keys for the output file.

ERROR in [ENV];ConvertVELfile: *Bad header for velocity file*

This message occurs when the velocity summary input file is empty.

ERROR in [ENV]:ConvertVELfile. *Bad header in velocity file.Field COUNT5 is missing*

This message, and other similar messages with different field names, occurs when there is a missing necessary field in the velocity file. The fields needed are `TIME, LINK, NODE, DISTANCE, COUNT0, COUNT1, COUNT2, COUNT3, COUNT4`, and `COUNT5`. If any of these fields are missing, the simulation must be rerun with the configuration modified to allow collection of all of these fields.

4.4.1.2 Messages and ERRORs from the *EmissionsEstimator*

Usage: `EmissionsEstimator <configFilename>`

This message occurs when the *EmissionsEstimator* is run without a configuration file as an argument on the command line. Specify the configuration file used to run the Traffic Microsimulator to produce the velocity data for the Emissions Estimator.

ERROR: *Network configuration keys* `NET_DIRECTORY, NET_NODE_TABLE`*, or* `NET_LINK_TABLE` *is not defined*

This version of the environmental model uses road network tables. These three keys must be set in the configuration file.

ERROR: *Unable to open the CA input file: <velocityFilename>*

This message occurs when the *EmissionsEstimator* program is unable to locate or open either of the postprocessed velocity input files. Make sure that the file is present in the directory in which you are attempting to run the *EmissionsEstimator* or that the configuration file keys `EMISSIONS_LDV_VELOCITY_FILE` and `EMISSIONS_HDV_VELOCITY_FILE` contains the correct file name.

ERROR: *Unable to open the composite array input file: arrayp.out*

This message occurs when the *EmissionsEstimator* program is unable to locate or open the composite array input file. Make sure that the *arrayp.out* file is present in the directory in which you are attempting to run *EmissionsEstimator* or that the configuration file key `EMISSIONS_COMPOSITE_INPUT_FILE` contains the correct file name.

ERROR: *Unable to open the composite differences array input file: arraypd.out*

This message occurs when the *EmissionsEstimator* program is unable to locate or open the composite array input file. Make sure that the *arrayp.out* file is present in the directory in which you are attempting to run *EmissionsEstimator* or that the configuration file key `EMISSIONS_COMPOSITE_DIFF_INPUT_FILE` contains the correct file name.

ERROR: *Unable to open the composite array parameter input file: ARRAYP.INP*

This message occurs when the *EmissionsEstimator* program is unable to locate or open the composite parameter input file. Make sure that the *ARRAYP.INP* file is present in the directory in which you are attempting to run the *EmissionsEstimator* or that the configuration file key `EMISSIONS_ARRAY_PARAMETERS_FILE` contains the correct file name.

ERROR: *Unable to open the vehicle distribution input file: vehcold.dis*

This message occurs when the *EmissionsEstimator* program is unable to locate or open the vehicle distribution input file. Make sure that the *vehcold.dis* file is present in the directory in which you are attempting to run the *EmissionsEstimator* or that the configuration file key `EMISSIONS_VEHICLE_COLD_DISTRIBUTION` contains the correct file name.

ERROR: *Unable to open the output file: emissions.ldv.out*

This message occurs when the *EmissionsEstimator* program is unable to open the output file. This may occur because of lack of disk space or lack of write privileges in the directory in which you are attempting to run the *EmissionsEstimator*. If not using the default output filename *emissions.ldv.out*, make sure the configuration file key `EMISSIONS_LDV_OUTPUT_FILE` contains the correct filename.

ERROR: *Unable to open the debugging output file: <debugFilename>*

This message occurs when the *EmissionsEstimator* program is unable to open the debugging files. This may occur because of lack of disk space or lack of write privileges in the directory in which you are attempting to run the *EmissionsEstimator*. If not using the default debug filename (*debug.out* and *calcsum*), make sure the configuration file keys `EMISSIONS_DEBUG_FILE` and `EMISSIONS_DEBUG2_FILE` contains the correct filenames.

ERROR: *Input file vehcold.dis does not have enough soak times*

This messages occurs when eight lines are not present in the *vehcold.dis* file. There must be a line for the number of soak time bins. If the default (*vehcold.dis*) filename is not used, check the `EMISSIONS_VEHICLE_COLD_DISTRIBUTION` configuration file key to verify the correct filename.

ERROR: *Input file ARRAYP.INP does not have enough data*

This messages occurs when ten data elements are not present in the *ARRAYP.INP* file or the file specified by the `EMISSIONS_ARRAY_PARAMETERS_FILE` configuration file key.

ERROR: *Missing a header line from input file: arrayp.out*

This messages occurs when the *arrayp.out* or *arraypd.out* (or files specified by the `EMISSIONS_COMPOSITE_INPUT_FILE` or `EMISSIONS_COMPOSITE_DIFF_INPUT_FILE` configuration file keys) files do not have at least two lines in the file.

ERROR: *Out of input data in file: arrayp.out*

This messages occurs when there is not enough data in the *arrayp.out* or *arraypd.out* files (or the file specified by `EMISSIONS_COMPOSITE_INPUT_FILE` or `EMISSIONS_COMPOSITE_DIFF_INPUT_FILE` configuration file key). It must contain six columns of data. The number of rows of data must be the maximum number of acceleration bins multiplied by the maximum number of velocity bins. These two values are read in from the *ARRAYP.INP* file. In this release of the emissions modules, these values are 20 speed bins and 34 power bins. There should be 680 rows of data in the *array.out* file.

ERROR: *Emissions Estimator : Calloc failure)*

This messages occurs when the *EmissionsEstimator* program was unable to allocate enough storage to do its calculations on a specific area of the network.

ERROR*: Missing data from input file: velocity.ldv.out*

This message occurs when the *velocity.ldv.out* file is missing data (or the file specified by the `EMISSIONS_LDV_VELOCITY_FILE` configuration file key). The value after *nv* = should be the number of lines of data following for that particular time, link, and node. Errors in the *velocity.ldv.out* file would have been introduced by the *ConvertVELfile* program.

WARNING*: Data set contains a link with empty interior boxes*

There must be vehicles present in a consecutive set of boxes on any particular link. Links may contain boxes that are empty at the beginning and the end of the link. In order to calculate emissions correctly, boxes that contain vehicles must all be consecutive. There may not be empty boxes in the middle of a set of boxes with vehicles. One way to check if a box contains vehicles is to look across a particular row of data in the *velocity.ldv.out* file. If all values are zero, then there were no vehicles in that box. Links that contain empty interior boxes will be discarded and emissions will not be calculated on them.

WARNING: *Data set contains a link with data in first box only*

This message occurs when there is a non-zero value in the first column of velocity counts and all other counts in that row are zero. The *EmissionsEstimator* program is unable to calculate emissions for this situation and, therefore, that link will be discarded from the output data.

WARNING: *NaN or Inf found in one of the output files (emissions.ldv.out, debug.out, or calcsum)*

This messages occurs when there is a calculation error in the *EmissionsEstimator* program (such as a divide by zero). No known fix for this.

4.5 Output Visualizer

4.5.1 Running the Output Visualizer

To run the Output Visualizer, type the following command line:

```
Vis configfile
```

where `Vis` is the executable file, and `configfile` is the name of the configuration file upon which you would like to view the data.

Do not run this application in the background because you will need a text output window to retrieve various information types.

4.5.2 Manipulating the View

4.5.2.1 Translating

To translate, perform the following four steps:

1) Click on [Translate].

2) Use the left mouse button to click a point in the viewing area.

3) Hold the button down and drag to another point in the viewing area.

4) Release the left mouse button.

The Output Visualizer redraws the network, which is translated by the distance between the two points.

To undo a translation, click [Default View], which resets the viewing transformations to their original values and redraws the network.

4.5.2.2 Zooming

Zoom-in

To zoom-in, perform the following steps:

1) Click on [Zoom In].

2) Use the left mouse button to click a point in the viewing area.

3) Hold the button down and drag to another point in the viewing area.

4) Release the left-mouse button.

The Output Visualizer displays the selected area with the maximum magnification allowable in the monitor's window.

To undo a zoom operation, click [Default View], which resets the viewing transformations to their originals values and redraws the network. It also can be undone by clicking [Zoom Back].

Zoom-back

Zoom-back reverses the last zoom-in operation. Up to 50 zoom-in operations can be undone.

Note that when selecting an area to zoom into, any diagonal in any direction will work. For example, the same results can be produced by clicking on the upper left then dragging to the lower right of an area, as when clicking on the lower right and dragging to the upper left.

Fig. 14 provides a sample zoomed-in view of the network. Note that, in this figure, the lane dividers are displayed as white dashed lines and that the yellow lines are drawn on the left side of the links, thereby indicating the direction of traffic.

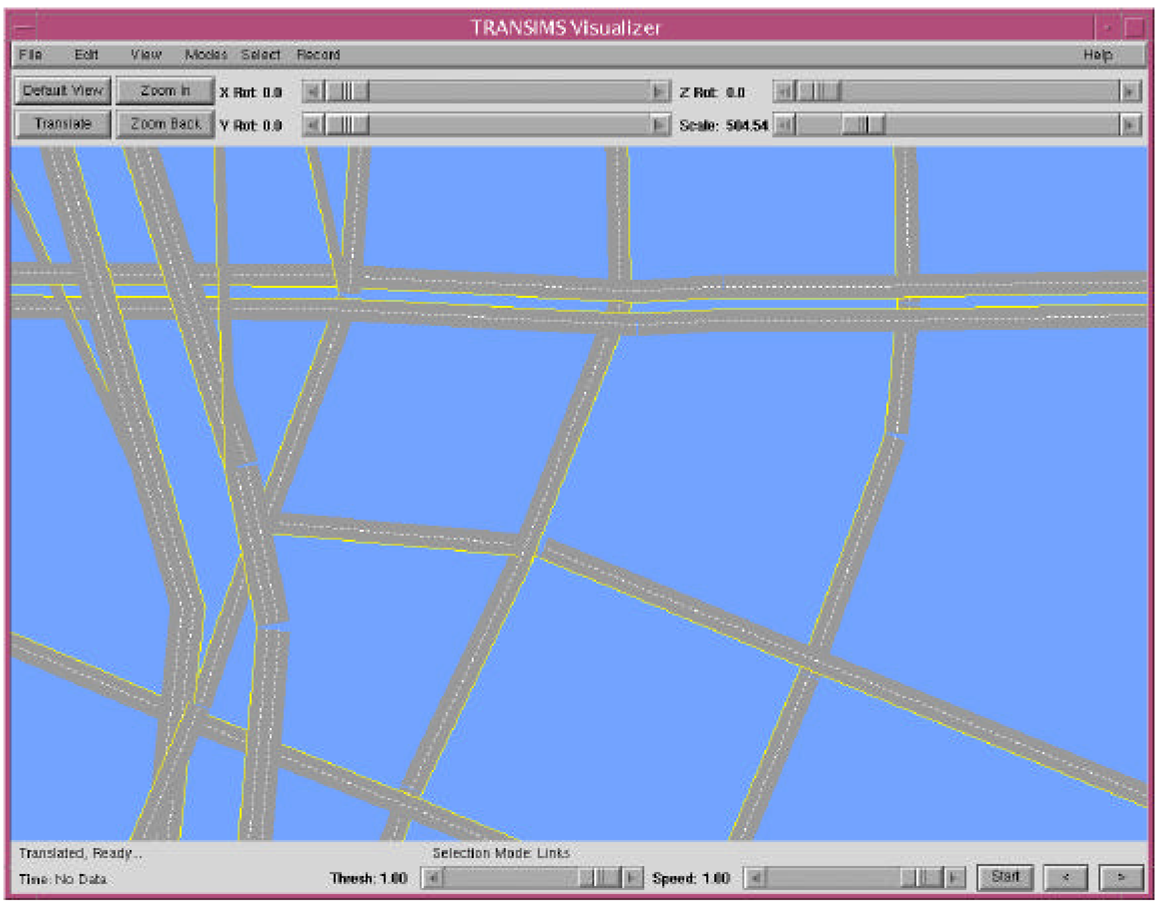


Fig. 14. The Output Visualizer displaying a zoomed-in view of the Portland EMME/2 Network.

4.5.2.3 Rotating

To rotate an object, drag the thumb button to the right of the “Z Rot” label. The entire network will rotate about the Z-axis by the amount shown on the slider. You can also click on the arrows at the ends of the slider to increase or decrease the rotation.

4.5.2.4 Scaling

To scale, drag the thumb button to the right of the “Scale Label.” The entire network will be magnified about the current center of the window by the amount shown by the slider. You may also click on the arrows at the ends of the slider to increase or decrease the scale.

4.5.2.5 Displaying Vehicle Evolution Files

To create an Indexed Vehicle Evolution File, use the *indexvehtobin* utility to convert a vehicle-snapshot file into a binary format (see Chapter 6 for a description of this utility). Once this is done, read in a Vehicle Evolution File by selecting File→Open Indexed Vehicles.

The text in the status bar changes to *Reading in Vehicles*… while the vehicle evolution file is being read and will return to *Ready*… once completed.

The Output Visualizer displays the vehicles as points when zoomed out and transitions to displaying vehicles as triangles when zoomed in. The display should now look similar to that shown in Fig. 15.

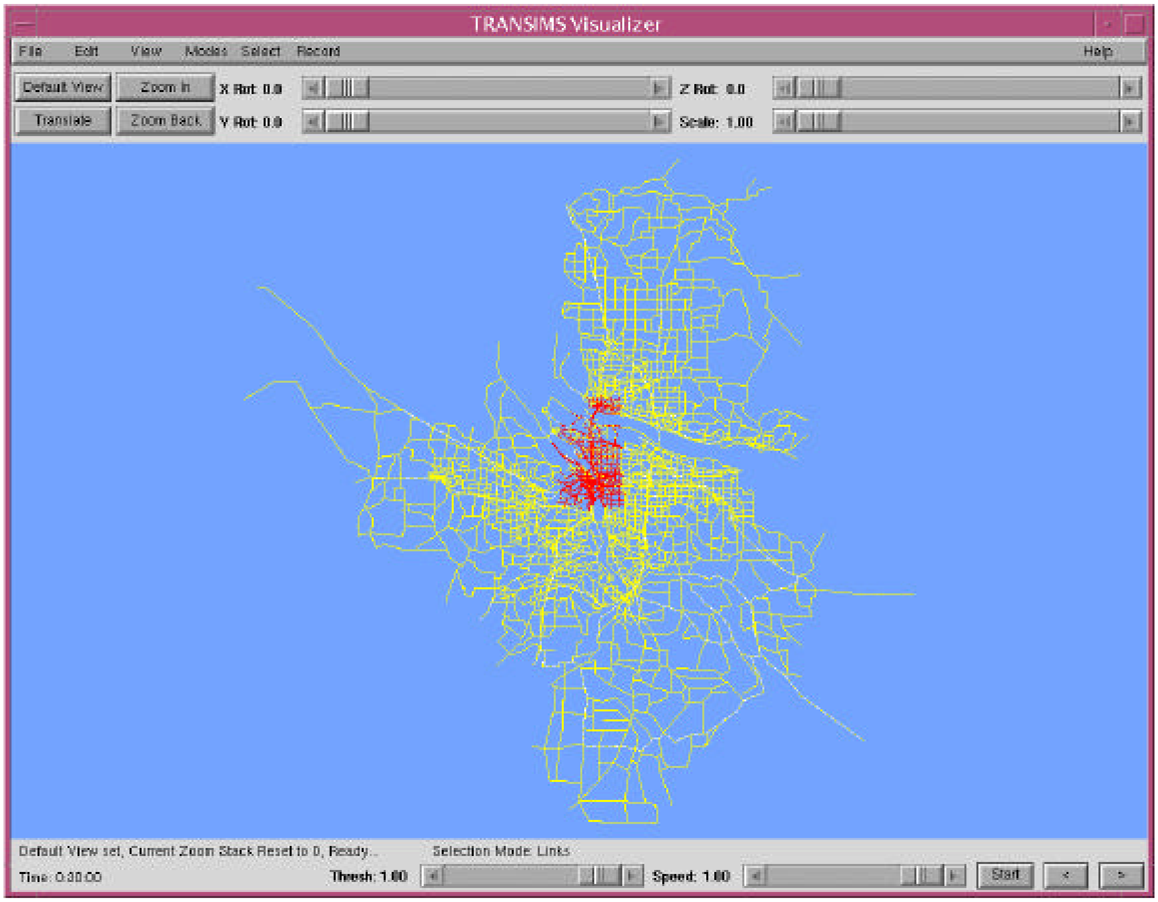


Fig. 15. The Output Visualizer displaying vehicles on the Portland EMME/2 Network.

4.5.2.6 Looking for High Vehicle Densities

The leftmost slider at the bottom of the window is labeled "Thresh." This slider controls the transparency of the vehicles when vehicles are shown as points. When Thresh is 1.00, the vehicles are not transparent; as thresh declines, they become more transparent.

Thus, because in very zoomed-out views vehicles are smaller than pixels, pixels with multiple transparent vehicles will appear darker. The highest densities of vehicles will appear as the darkest pixels as the Thresh slider goes towards zero (see Fig. 16).

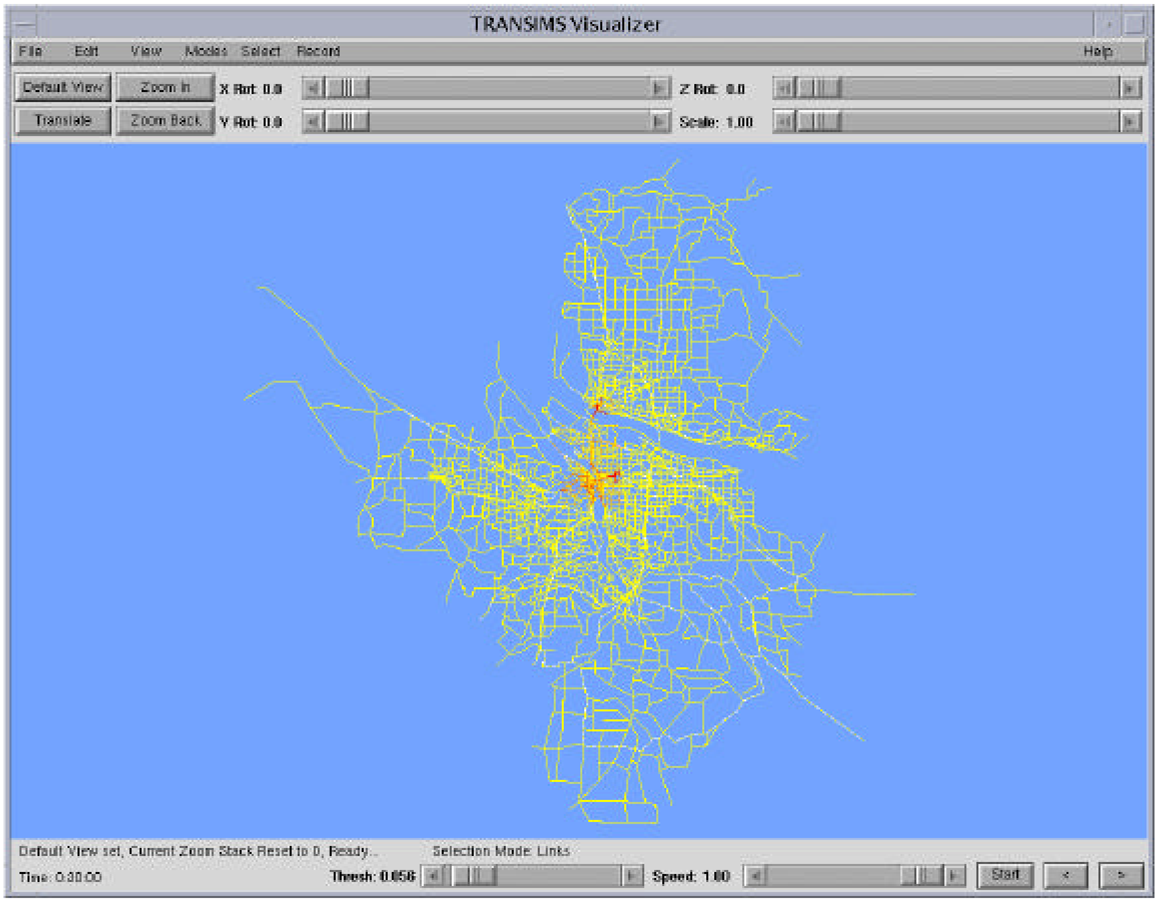


Fig. 16. The Output Visualizer displaying high vehicle densities on the Portland EMME/2 Network.

If you zoom in, the vehicles will be displayed as triangles (see Fig. 17). The point of the triangle in the middle of the lane represents the front of the vehicle.

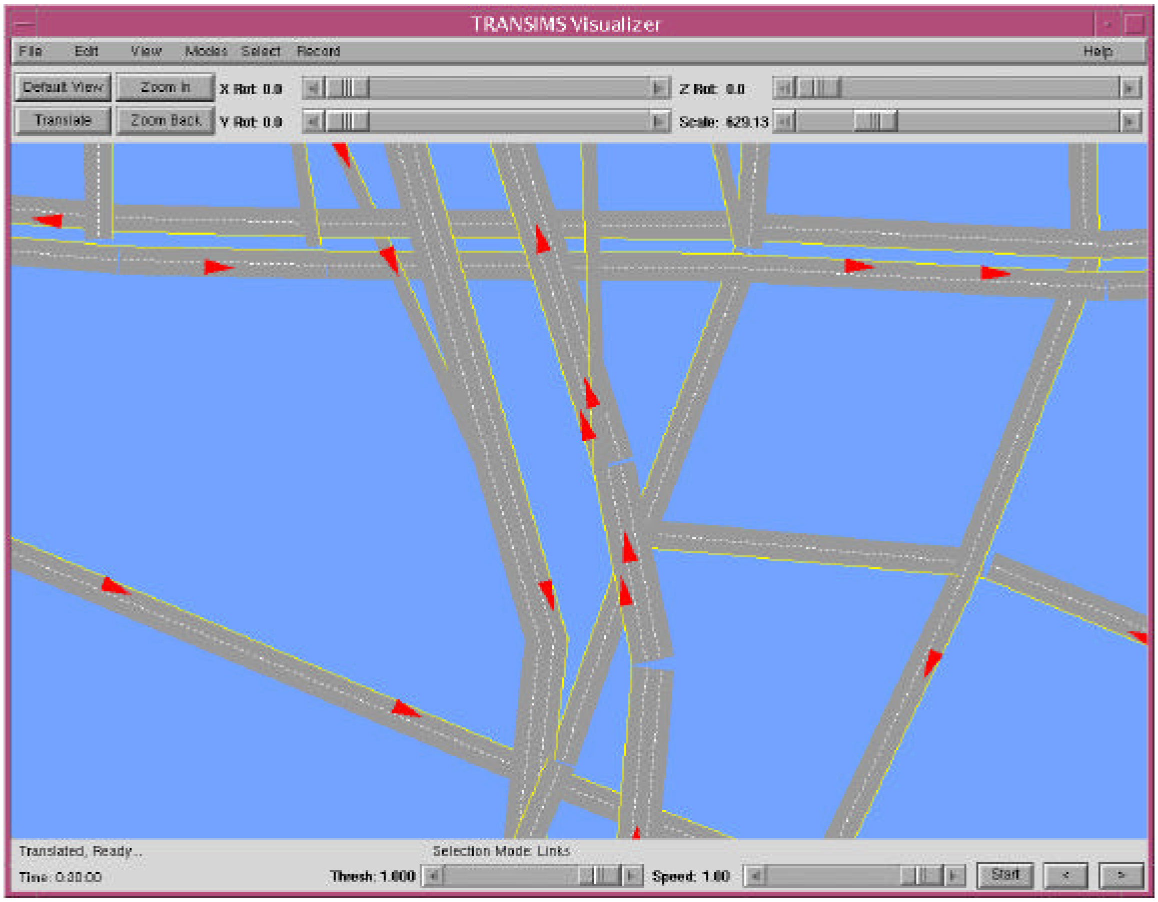


Fig. 17. The Output Visualizer displaying vehicles on the Portland EMME/2 Network by using a zoomed-in view.

4.5.2.7 Coloring the Vehicles by Different Schemes

By selecting View→Vehicles, you will call up the Select and Vehicle Viewing Parameters dialog box. This dialog box enables you to select a scheme by which to color the vehicles. You may

- have the vehicles all appear in one color,
- color the vehicles by type,
- color by number of passengers,
- color by velocity,
- color by user field, or
- color the vehicles with random colors.

Coloring the cars randomly proves useful when watching the flow of large numbers of vehicles and yet retaining the focus on a particular vehicle. Different vehicle types can be distinguished by coloring by Type.

Vehicles can also be drawn in 3-D in any mode by selecting the 3-D Vehicles checkbox at the bottom of the Vehicle Viewing Parameters dialog box. To accomplish this, select Modes→Color by Velocity, and click [OK]. The vehicles are colored according to velocity: dark red (being the slowest), then bright red, then blue, then bright green, followed by dark green (for the fastest vehicles).

4.5.2.8 Animating the Vehicles

Perform the following steps to animate the vehicles:

1) To start the animation, click the left mouse button on the viewing area. As each animation frame is drawn, the timestep is written in the status bar.
2) To stop the animation, click the right mouse button.
3) To slow the animation speed, drag the Speed slider towards “0” (left).
4) To display the starting timestep, click [Start].
5) To advance the timestep, click [>]. To decrease it, click [<].

Animating Faster

Because the entire network is redrawn for each frame, the animation may appear quite slow, particularly for networks with large numbers of links. The animation speed can be greatly increased by switching on the Overlay mode. This is done by selecting Modes→Overlay.

Once this mode is on, restart the animation. The animation frame rate now should be much faster because the network is drawn only once and is copied from memory into the viewing area for each subsequent timestep. Note that the Overlay mode is useful only for larger size networks (more than 3,000 links), as smaller networks often can be drawn faster than the pixel transfer can take place.

4.5.2.9 Retrieving Information about a Vehicle

To retrieve information about a certain vehicle, click on the front middle of it (when it is large enough to be represented as a triangle) with the middle mouse button. If the vehicles are represented by points, click on a point within five meters of the middle front of the vehicle. This in turn will bring up information in the text window from which you started the Output Visualizer and on the selection status line. A typical output is as follows:

Found Vehicle: ID 336565 Type 16 Passengers 0
Velocity 24.59 Link ID 16155 User: 0

Because the first vehicle whose point is within the five-meter tolerance is selected for output, try to select a point in which it is not likely that the point will be within the tolerances of another vehicle. If a vehicle cannot be found at the point where you have clicked, the following message is displayed:

Vehicle NOT FOUND at xcoordinate ycoordinate

4.5.2.10 Ride-in-Vehicle Mode

Sometimes it is convenient to have the viewpoint set as if you were riding in a given vehicle. To accomplish this, use the Modes→Ride in Vehicle Mode menu option.

To use this mode, perform the following steps:

1) Select a vehicle ID and enter it into the dialog box that is displayed when Edit→Find Vehicle is selected.
2) Select the Modes→Ride in Vehicle Mode menu item, and the entire data set will be searched for the chosen vehicle ID and processed into a new data set.
3) Check to make sure there are no warnings (there is at least one record for the chosen vehicle ID); the viewpoint will be set as if you were riding in that vehicle.

Notice that the text labeling on the sliders has changed to Roll, Pitch, Yaw, and Height. These sliders can be used to manipulate the direction you are looking in (much like the controls of an airplane). Fig. 18 provides a sample view created in this mode.

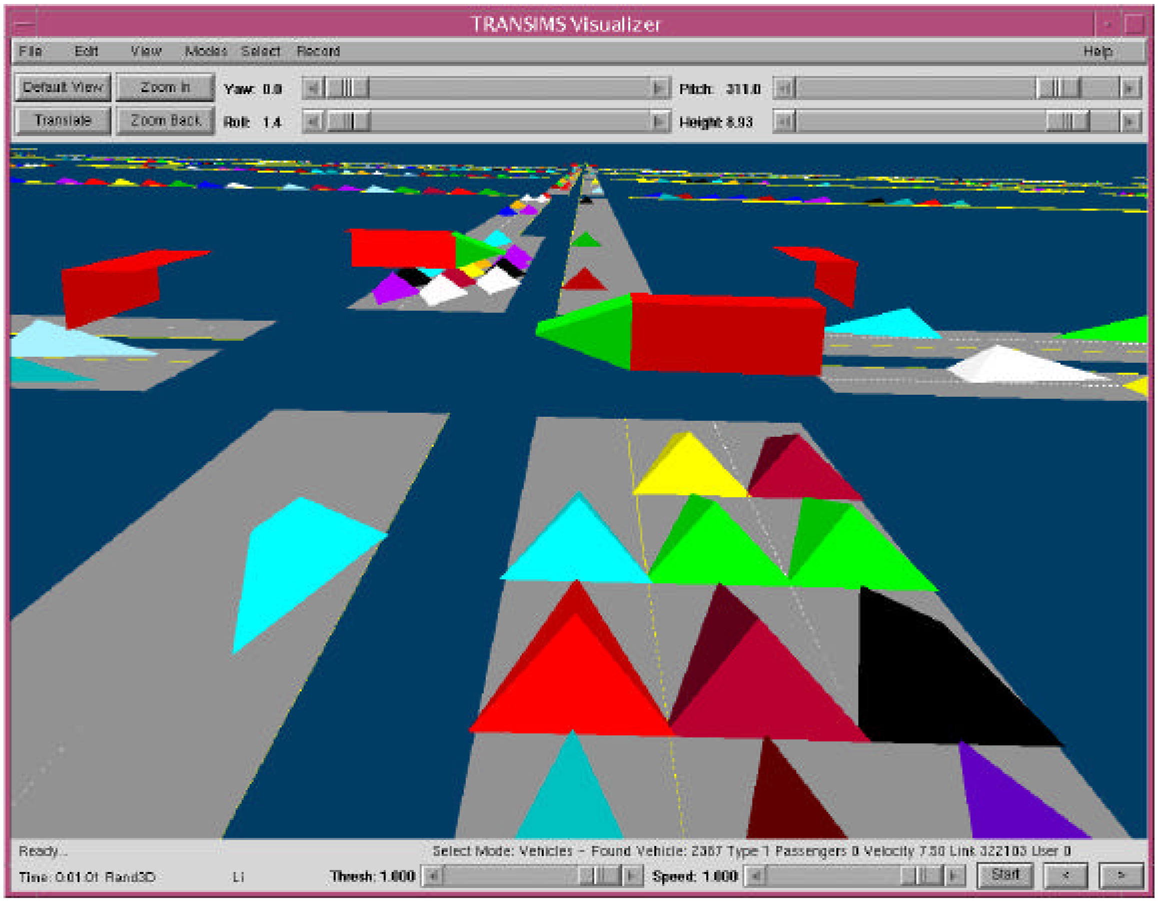


Fig. 18. The Output Visualizer displaying a view in the Ride-In Vehicle mode.

To toggle back to the default viewpoint, select the menu item again. When you are ready to go on to viewing another type of data, stop the animation and free the memory used for the vehicle evolution data by selecting File→Close Indexed Vehicles.

4.5.2.11 Viewing Variable-Size Box Data

To read in a Variable-Size Box data file, select File→Open Variable Size Box. The text in the status bar will change to *Reading in Variable Size Boxes…* while the file is being read and will return to *Ready…* once completed.

The Output Visualizer displays the first column of data in the file by default. Links are colored according to the user-selectable colormap.

Other columns can be displayed by selecting View→Increment Data Column, which increments the data column being displayed. Data animation can be displayed by clicking the left mouse button in the viewing area; stopping the animation is done by clicking the right mouse button.

Variable-Size Box data files can be produced from plans by using the *Plan2BoxSummary* utility. The emissions package also produces files of this type.

4.5.2.12 Viewing the Data in 3-D

Because it can be very difficult to identify outlying values with colormapped data, the Output Visualizer provides 3-D capabilities. It is much easier to identify outlying data when viewing tall bar graphs than by coloring the data values with a given colormap.

To view data in 3-D, perform the following steps:

1) Select View→Variable Size Boxes. A dialog box is displayed that allows you to enter a scale factor for multiplying the data values into 3-D bar heights.

2) Enter an appropriate scale factor, and click [OK].
The display is rendered with 3-D bars on the boxes.

3) Rotate the display to approximately 290 degrees with the X Rot slider.

4) Left-click to animate the data.

5) Right-click to stop the animation.

The display may look confusing at this point because all faces of the 3-D bar boxes are colored exactly the same. You will not be able to distinguish the tops of the boxes from the sides.

The lighting model fixes this problem. Switch on the lighting model by selecting Modes→Lights On/Off. Fig. 19 provides a sample view of 3-D Variable-Size Box data.

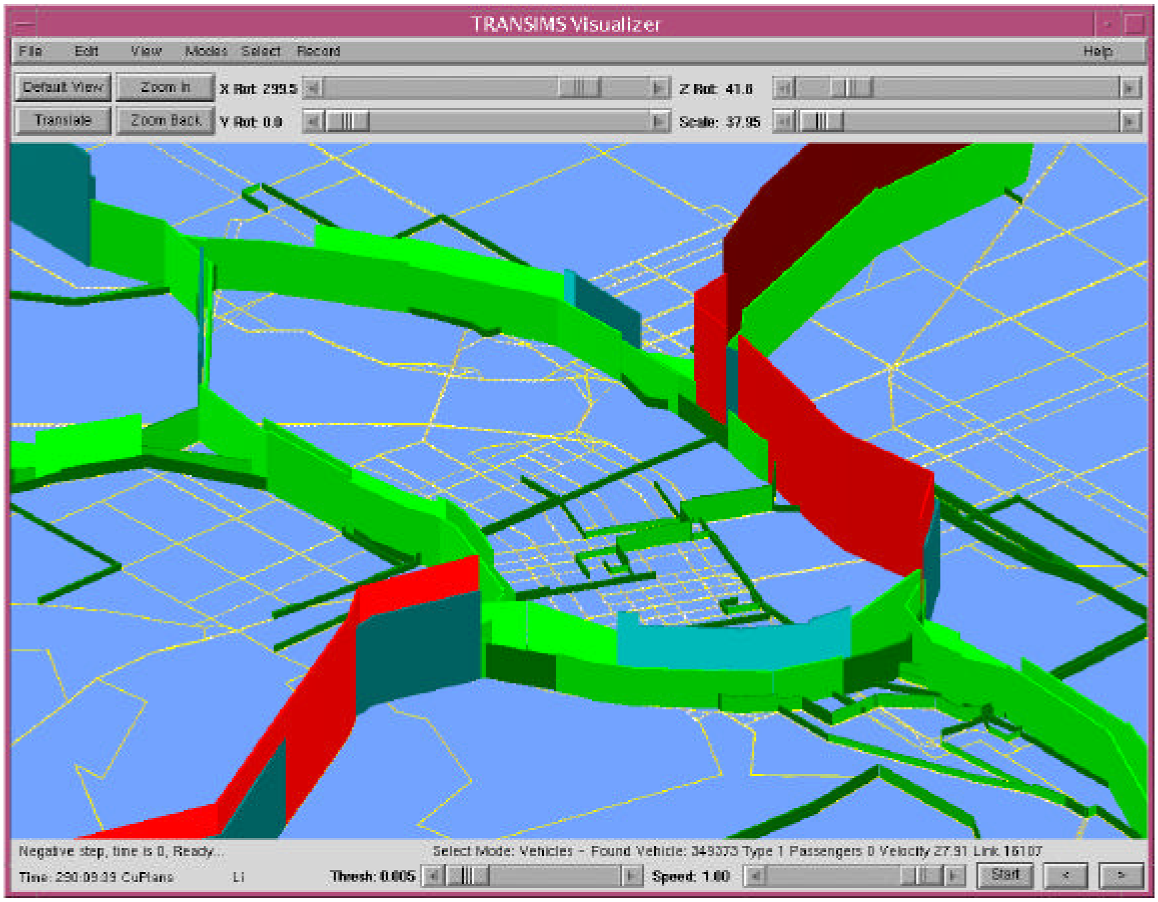


Fig. 19. The Output Visualizer displaying cumulative plan data.

4.5.2.13 Varying the Colormap Thresholds

In some cases, it may be desirable to determine whether the data value of one link is higher than another. Because of the colormap, however, the two links will have the same color.

Changing the colormap into a more finely grained colormap with more colors can alleviate this problem, but there is an alternative and more convenient way. By cycling the color map (that is, by increasing the minimum value color), the links on the display change colors.

The link with the higher data value associated with it will change colors before the other link of the same initial color. This cycling of colormaps is accomplished by using the Thresh slider.

Free the memory used for the Variable-Size Box data by selecting File→Close Variable Size Box Data.

Appendix A: Bignet Systematic Number System

Systematic Numbering of IDs

Because there are so many links in the Bignet Network, it is very difficult to diagnose problems and find exactly where a certain error took place. To alleviate this problem, we have systematically numbered all objects (e.g., links, nodes, parking, and activity locations) so that the user can find an object's type and location by using its unique ID.

This numbering scheme is used for all of Bignet except for the numbering of freeway, bridge, and rail-only links. This numbering scheme is an independent system. Thus, we have divided the remainder of this discussion into two sections:

1) the first addresses the majority of the Bignet numbering scheme, and

2) the second discusses the numbering scheme for freeway, bridge, and rail-only links.

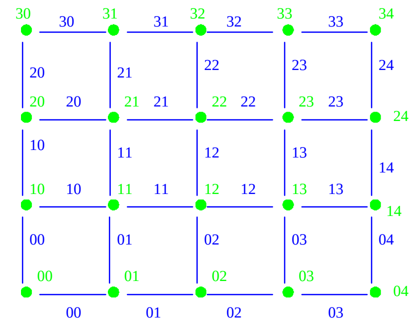
Numbering Scheme for the Majority of Bignet

Every ID consists of five to seven digits, each of which has a special meaning. Of these digits, all but the last four depend on the object type. The last four digits give the object's location. In this four-digit sequence, the first two specify the block group location and the last two specify the location within that block group (this is shown in Fig. 20).

As shown in Fig. 20, the southwest-most local block group has a block ID of 11 (this is not "eleven," but rather "one-one"—remember, every digit means something). The next block to the north is block 21, and the next block group to the east of 11 has ID 12. Note that the first digit is the y-block and the second is the x-block. Every object in Bignet has these two digits as fourth and third from last. For example, a node in the west half of the downtown area will have an ID of x35xx. The "x" symbols are explained in the following paragraph.

The last two digits of every ID denote the location within the block, again numbered from the southwest corner. Each node has a unique location. As shown in Fig. 20, the southwest-most node had local ID of 01 (zero-one). Thus, if these nodes were in the aforementioned downtown block group, they would have IDs x3500, x3510, and x3501. The "x" denotes the node type.

Block group numbering (YX)



Link and node numbering (yx)

Fig. 20. ID numbering in Bignet.

Because each node can be specified by only the last four digits, adding another digit is not expressly necessary, but it does help the user diagnose problems. The different node types are detailed in Table 26. Links are numbered relative to the nodes to which they are connected. A link ID's last four digits match those of the node to the south or the west-end of the link. (All links are either north-south or east-west.)

Table 26. Node control types.

Node type	ID	Notes
Seamless	1	Freeway-Primary Arterial connection
Signalized	2	Signalized intersection with left turn pockets
4-Way stop	3	
NS Stop	4	NS road must stop, EW road has no control
EW Stop	5	EW road must stop, NS road has no control
NS Yield	6	NS road must yield, EW road has no control
EW Yield	7	EW road must yield, NS road has no control
Endnote	8	End of network (no more links)

Unlike nodes, a link location does not uniquely specify the link, because two links will have the same four location digits. Therefore, the fifth digit from the right specifies the orientation of the link relative to the node with the same four location digits (see Table 27).

Table 27. Link orientation from given node.

ID	Meaning
1	Link is north of node having same location
2	Link is east of node having same location

As with nodes, each link ID includes one more piece of information beyond that required for unique specification: the link type. Listed in Table 28, these types have been discussed in previous sections. To summarize link IDs, this information is organized as "`tdYXyx`," where

- `t` denotes the link type in Table 28,
- `d` corresponds to the direction in Table 27,
- `Y` represents the north-south block in which the link is located,
- `X` is the east-west block,
- `y` is the north-south local location within that block, and
- `x` is the east-west local location within that block.

Again, note that "`Yxyx`" corresponds uniquely to the node east or south of the link.

Table 28. Link types.

Link type	ID	Notes
Freeway	5	two lanes in each direction, 5 cells per second
Primary Arterial	4	two lanes in each direction, 4 cells per second
Secondary Arterial	3	two lanes in each direction, 3 sells per second
Collector	2	one lane in each direction, 2 cells per second
Local	1	one lane in each direction, 2 cells per second

As discussed earlier, most links have parking locations and activity locations, both of which have some transit stops on some secondary arterials. These are all classified as features on the link. Although physically part of a link, pocket lanes are also considered to be a link feature in TRANSIMS. Moreover, all activity locations must be connected to other link features using process links. All of these objects share the link's four location digits. Like links, they all have six digits, the first of which is the feature type (listed in Table 29), and the second of which lists a direction or orientation.

Table 29. Link feature designations.

Feature type	ID	Notes
Pocket	6	for left turns at signalized intersections
Parking	7	one on each side of link
Activity	8	only one per link
Transit Stop	4	one on each side of link
Parking-Process	9	transfer between Parking and Activity Location
Stop-Process	5	transfer between Transit Stop and Activity Location

The orientation digit will be different for distinct types of features. Pocket lanes, parking locations, and transit stops all have the orientation digit filled to identify the direction of the lane to which they are attached. Because they can be on either side of the link and in

Bignet links are bi-directional, this includes all four compass directions listed in Table 30.

Table 30. Pocket, parking, and transit stop orientations.

Direction	ID
North	3
South	4
East	5
West	6

Although parking lots are considered to be on one side of the link, they can be accessed from traffic in either direction on the link. Transit stops can be accessed only from traffic moving on the same side of the road. We need this information so we know the end of the link the left-turn lane occupies on pocket lanes.

Process links require a little more information encoded in the digit because each activity location must be connected to parking and transit stops in both directions. Furthermore, process links are unidirectional (and thus there are twice as many possible values). Table 31 lists the possible values of this digit. We do not need to distinguish whether it is a parking lot or transit stop connected to the activity because that has already been done in the ID's first digit (see Table 30).

Table 31. Process link direction s(X = parking location or transit stop).

ID	Meaning
1	from X to Activity on northbound part of link
2	from X to Activity on southbound part of link
3	from X to Activity on eastbound part of link
4	from X to Activity on westbound part of link
5	to X from Activity on northbound part of link
6	to X from Activity on southbound part of link
7	to X from Activity on eastbound part of link
8	to X from Activity on westbound part of link

Each link that has activity locations has exactly two—one on each side of the road. In Bignet, there is at most one parking lot on each link; the lot is connected to both activity locations with "from" and "to" process links.

If a Bignet link has transit stops, it has exactly two—one on each side of the road. Each transit stop is connected only to the activity location on its own side of the road. If an individual gets off transit on the northbound side, that individual must walk to the intersection and cross to get to the southbound-side activity location.

Freeway, Bridges, and Rail-only Links

Bridges are labeled in a way that is similar to the rest of the network, except that

- the links at the beginning and the end of the block have been stretched to connect, and

- all the links in between these two have been deleted.

The numbering of freeway and rail-only links is in no way systematic, except that all objects associated with them have ID values less than 100.

Fig. 21 shows a subset of the links and nodes on both sides of the river. The figure includes the bridges, the freeway, and two of the three rail-only links. Colors are defined as follows:

- nodes are red,
- links are black,
- freeway links are green, and
- rail-only links are in blue.

Note that the freeway has connections to nodes only at the merge and turn lanes (nodes 2, 4 and 7).

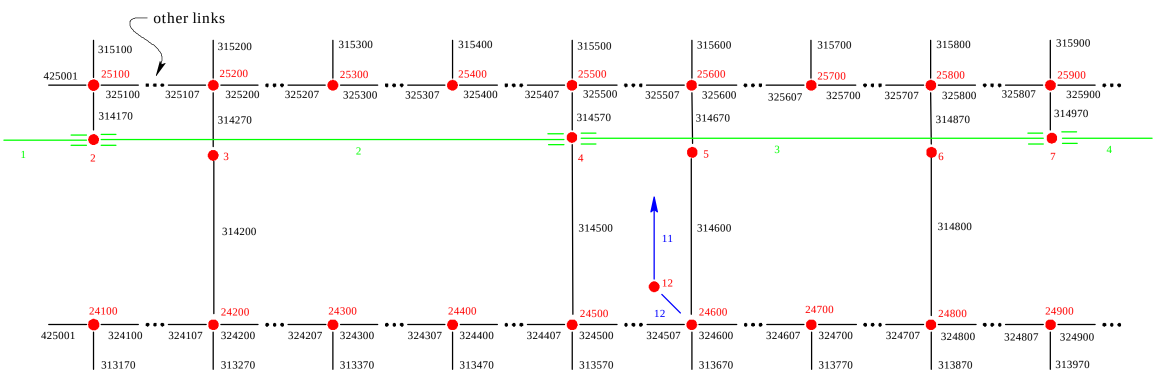


Fig. 21. Link and node numbering around the river in Bignet (not to scale).

The rail line consists of three rail-only links:

- At 9,185 meters in length, the first link extends from just outside the local black groups to the northeast corner of block 86.
- At 7,200 meters in length, the second link runs from block 86 to just northwest of the intersection at node 24600.
- At 60 meters in length, the third links connect the second link to that node, and thus it connects to local streets.

All three links have one lane in each direction and a maximum speed of five cells per second (37.5 m/s). Because there are no signals or competing traffic, the rail lines tend to run at this speed when on the rail-only links. On the local streets, they must obey the slower speed limit and compete with other vehicles. The rail-only links are depicted in Fig. 22 This figure shows only the secondary arterials for reference.

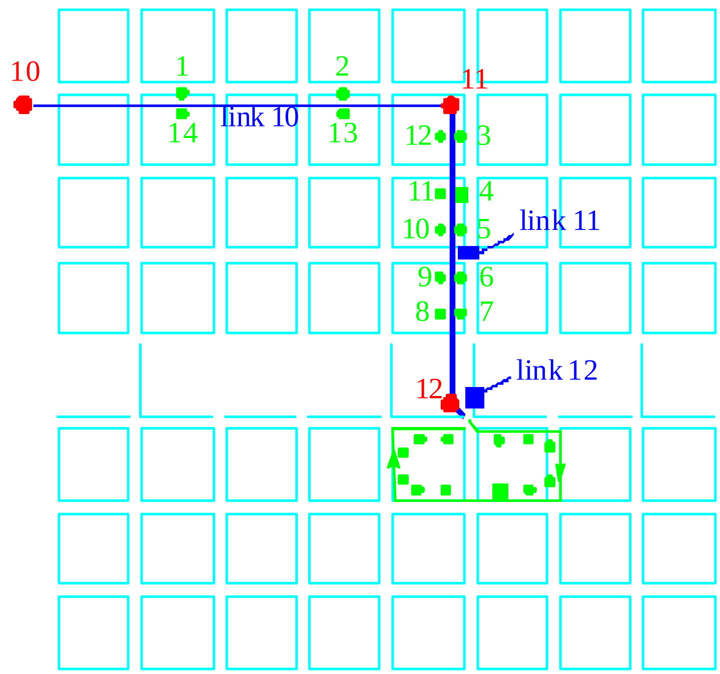


Fig. 22. Numbering on the rail-only links (nodes in red, links in blue, and stops in green).

There are two stops in each direction on link 10, which are numbered 1, 2, 13, and 14. These are the first two and last two stops of the rail line, respectively. Stops 1 and 14 are connected to activity location 858202, and stops 2 and 13 are connected to activity location 858402 through process links. On link 11, there are five more stops in each direction, also connected to the nearest activity location in each case. For all stops, the activity location to which they are connected also happens to be connected to a bus stop.

Appendix B: Valid Person and Household Demographic Field Names

Following is a list of the demographic options that are available for households and the allowed values for each demographic item.

```
DATA           SIZE         BEGIN

D  RECTYPE          1            1
        Record Type
V         H  .Housing Record

D  SERIALNO         7            2
V 0000000..
    9999999  .Housing unit/GQ person serial number unique
             .identifier assigned within state or state group

D  SAMPLE           1             9
        Sample Identifier
V         1  .5% sample
V         2  .1% sample
V         3  .Elderly

D  DIVISION         1            10
        Division code
V         0  .Region/division not identifiable
V            .(Selected MSA/PMSAs on 1% sample)
V         1  .New England (Northeast region)
V         2  .Middle Atlantic (Northeast region)
V         3  .East North Central (Midwest region)
V         4  .West North Central (Midwest region)
V         5  .South Atlantic (South region)
V         6  .East South Central (South region)
V         7  .West South Central (South Region)
V         8  .Mountain (West region)
V         9  .Pacific (West region)

D  STATE            2            11
        State Code
V    01..56  .FIPS state code (See appendix I-59)
V        99  .PUMA boundaries cross state lines – 1% file

D  PUMA              5           13
        Public use microdata area (state dependent)
V   00100..
V     99999  .PUMA code (Includes tract groups) 1st 3
V            .Digits = main PUMA – generally county place
V            .Last 2 digits = groups of tracts, BNA, etc.

D  AREATYPE         2            18
        Area type revised for PUMS equivalency file
         (See Appendix C-1)
V        10  .Central city
V        11  .Central city part
V        20  .MSA/PMSA – Outside central city
V        21  .MSA/PMSA – Outside central city (part)
V        22  .Central City (part) & outside central city
V            .(part)
V        30  .Entire MSA
V        31  .2 or more MSAs/PMSAs
```

```
V        40  .Mixed MSA/PMSA/NON-MSA/PMSA area
V        50  .Outside MSA/PMSA
V        60  .Place
V        61  .Place – part
V        70  .MCDs/Towns (New England only)
V        80  .Counties/independent Cities (2 or more)
V        81  .County/independent city – part
V        82  .County/independent city

D  MSAPMSA          4           20
        MSA/PMSA
V    0040..
V      9360  .FIPS/MSA/PMSA code, selected MSA/PMSA
V            .(See appendix G)
V      9997  .Mixed MSA/PMSA NONMSA/PMSA area
V      9998  .2 or more MSAs
V      9999  .Not in MA

D  PSA              3           24
        Planning service area (elderly sample only –
         state dependent)
V       000  .N/A (Elderly PUMS only)
V  1..18B  .Planning service area codes (See appendix G)

D  SUBSAMPL         2           27
        Subsample number (Use to pull extracts – 1/1000/etc.)
V    00..99  .See text. pp 4-45.

D  HOUSWGT          4           29
        Housing Weight
V    0000..
V      1152  .Integer weight of housing unit

D  PERSONS          2           33
        Number of person records following this housing
         record
V        00  .Vacant unit
V        01  .One person record (one person in household
V            .or any person in group quarters)
V    02..29  .Number of person records (number of persons
V            .in household)

D  GQINST           1           35
        Group quarters institution
V         0  .N/A (housing unit)
V         1  .Institutionalized
V         2  .Not institutionalized

D  HFILLER          3           36
        Filler

D  UNITS1           2           39
        Units in structure
V        00  .N/A (GQ)
V        01  .Mobile home or trailer
V        02  .One-family house detached
V        03  .One-family house attached
V        04  .2 Apartments
V        05  .3-4 Apartments
V        06  .5-9 Apartments
V        07  .10-19 Apartments
V        08  .20-49 Apartments
V        09  .50 or more apartments
```

```
V        10  .Other

D  HUSFLAG          1            41
        All 100% housing unit data substituted
V         0  .No
V         1  .Yes

D  PDSFLAG          1            42
        All 100% person data substituted
V         0  .No
V         1  .Yes

D  ROOMS            1            43
        Rooms
V         0  .N/A (GQ)
V         1  .1 Room
V         2  .2 Rooms
V         3  .3 Rooms
V         4  .4 Rooms
V         5  .5 Rooms
V         6  .6 Rooms
V         7  .7 Rooms
V         8  .8 Rooms
V         9  .9 or more rooms

D  TENURE           1            44
        Tenure
V             0  .N/A (GQ/vacant)
V         1  .Owned with mortgage or loan
V         2  .Owned free and clear
V         3  .Rented for cash rent
V         4  .No cash rent

D  ACRE10           1            45
        On ten acres or more
V         0  .N/A (GQ/not a one-family house or mobile home)
V         1  .House on ten or more acres
V         2  .House on less than ten acres

D  COMMUSE          1            46
        Business or medical office on property
V         0  .N/A (GQ/not a one-family house or mobile home)
          1  .Yes
V         2  .No

D  VALUE            2            47
        Property value
V        00  .N/A (GQ/rental unit/vacant, not for sale only)
V        01  .Less than $ 10000
V        02  .$ 10000 - $ 14999
V        03  .$ 15000 - $ 19999
V        04  .$ 20000 - $ 24999
V        05  .$ 25000 - $ 29999
V        06  .$ 30000 - $ 34999
V        07  .$ 35000 - $ 39999
V        08  .$ 40000 - $ 44999
V        09  .$ 45000 - $ 49999
V        10  .$ 50000 - $ 54999
V        11  .$ 55000 - $ 59999
V        12  .$ 60000 - $ 64999
V        13  .$ 65000 - $ 69999
V        14  .$ 70000 - $ 74999
V        15  .$ 75000 - $ 79999
```

```
V       16  .$ 80000 - $ 89999
V       17  .$ 90000 - $ 99999
V       18  .$100000 - $124999
V       19  .$125000 - $149999
V       20  .$150000 - $174999
V       21  .$175000 - $199999
V       22  .$200000 - $249999
V       23  .$250000 - $299999
V       24  .$300000 - $399999
V       25  .$400000 or more

D  RENT1           2           49
        Monthly rent
V       00  .N/A (GQ/not a rental unit)
V       01  .Less than $ 80
V       02  .$  80 - $ 99
V       03  .$ 100 - $124
V       04  .$ 125 - $149
V       05  .$ 150 - $174
V       06  .$ 175 - $199
V       07  .$ 200 - $224
V       08  .$ 225 - $249
V       09  .$ 250 - $274
V       10  .$ 275 - $299
V       11  .$ 300 - $324
V       12  .$ 325 - $349
V       13  .$ 350 - $374
V       14  .$ 375 - $399
V       15  .$ 400 - $424
V       16  .$ 425 - $449
V       17  .$ 450 - $474
V       18  .$ 475 - $499
V       19  .$ 500 - $524
V       20  .$ 525 - $549
V       21  .$ 550 - $599
V       22  .$ 600 - $649
V       23  .$ 650 - $699
V       24  .$ 700 - $749
V       25  .$ 750 - $999
V       26  .$1000 or more
V       27  .No cash rent (NCR)

D  MEALS           1           51
        Meals included in rent
V        0  .N/A (GQ/not a rental unit/rental-NCR)
V        1  .Yes
V        2  .No

D  VACANCY1        1           52
        Vacant usual home elsewhere (UHE)
V        0  .N/A (occupied or regular vacant/GQ)
V        1  .Vacant UHE-owner
V        2  .Vacant UHE-renter
V        3  .Vacant UHE-undetermined

D  VACANCY2        1           53
        Vacancy status
V        0  .N/A (occupied/GQ)
V        1  .For rent
V        2  .For sale only
V        3  .Rented or sold, not occupied
V        4  .For seasonal/recreational/occasional use
V        5  .For migratory workers
```

```
V        6  .Other vacant

D  VACANCY3         1            54
        Boarded up status
V        0  .N/A (occupied/GQ)
V        1  .Yes
V        2  .No

D  VACANCY4         1            55
        Months vacant
V        0  .N/A (occupied/GQ)
V        1  .Less than 1 month
V        2  .1 up to 2 months
V        3  .2 up to 6 months
V        4  .6 up to 12 months
V        5  .12 up to 24 months
V        6  .24 or more months

D  YRMOVED          1            56
        When moved into this house or apartment
V        0  .N/A (GQ/vacant)
V        1  .1989 or 1990
V        2  .1985 to 1988
V        3  .1980 to 1984
V        4  .1970 to 1979
V        5  .1960 to 1969
V        6  .1959 or earlier

D  BEDROOMS         1            57
        Bedrooms
V        0  .N/A (GQ)
V        1  .No bedrooms
V        2  .1 Bedroom
V        3  .2 Bedrooms
V        4  .3 Bedrooms
V        5  .4 Bedrooms
V        6  .5 or more bedrooms

D  PLUMBING         1            58
        Complete plumbing facilities
V        0  .N/A (GQ)
V        1  .Yes, all three facilities
V        2  .No

D  KITCHEN          1            59
            Complete kitchen facilities
V        0  .N/A (GQ)
V        1  .Yes
V        2  .No

D  TELEPHON         1            60
        Telephone in Unit
V        0  .N/A (GQ/vacant)
V        1  .Yes
V        2  .No

D  AUTOS            1            61
        Vehicles (1 ton or less) available
V        0  .N/A (GQ/vacant)
V        1  .No vehicles
V        2  .1 vehicle
V        3  .2 vehicles
V        4  .3 vehicles
```

```
V         5  .4 Vehicles
V         6  .5 Vehicles
V         7  .6 Vehicles
V         8  .7 or more vehicles

D  FUELHEAT         1            62
        House heating fuel
V         0  .N/A (GQ/vacant)
V         1  .Gas:  Underground pipes
V         2  .Gas:  Bottled, tank, or LP
V         3  .Electricity
V         4  .Fuel oil, kerosene, etc.
V         5  .Coal or coke
V         6  .Wood
V         7  .Solar energy
V         8  .Other fuel
V         9  .No fuel used

D  WATER            1            63
        Source of water
V         0  .N/A (GQ)
V         1  .Public system or private company
V         2  .Individual drilled well
V         3  .Individual dug well
V         4  .Other source such as a spring, creek, etc.

D  SEWAGE           1            64
        Sewage disposal
V         0  .N/A (GQ)
V         1  .Public sewer
V         2  .Septic tank or cesspool
V         3  .Other means

D  YRBUILT          1            65
        When structure first built
V         0  .N/A (GQ)
V         1  .1989 or 1990
V         2  .1985 to 1988
V         3  .1980 to 1984
V         4  .1970 to 1979
V         5  .1960 to 1969
V         6  .1950 to 1959
V         7  .1940 to 1949
V         8  .1939 or earlier

D  CONDO            1            66
        House or apartment part of condominium
V         0  .N/A (GQ)
V         1  .Yes
V         2  .No

D  ONEACRE          1            67
        House on less than 1 acre
V         0  .N/A (GQ, two or more units in structure)
V             1  .Yes
V         2  .No

D AGSALES           1            68
        1989 Sales of Agriculture Products
V         0  .N/A (less than 1 acre/GQ/vacant/
V            .2  or more units in structure)
V         1  .None
V         2  .$1 to $999
```

```
V         3  .$1,000 to $2,499
V         4  .$2,500 to $4,999
V         5  .$5,000 to $9,999
V         6  .$10,000 or more

D  ELECCOST          4            69
        Electricity (yearly cost)*
V       0000  .N/A (GQ/vacant)
V       0001  .Included in rent or in condo fee
V       0002  .No charge or electricity not used
V     0003..
        3099  .$3 to $3,099
V       3100  .Topcode
V      3101+  .$3101 or more = state median of topcoded
              .values

D  GASCOST           4            73
        Gas (yearly cost)*
V       0000  .N/A (GQ/vacant)
V       0001  .Included in rent or in condo fee
V       0002  .No charge or gas not used
V     0003..
        2099  .$3 to $2,099
V       2100  .Topcode
V      2101+  .$2101 or more = state median of topcoded
              .values

D  WATRCOST          4            77
        Water (yearly cost)
V        000  .N/A (GQ/vacant)
V        001  .Included in rent or in condo fee
V        002  .No charge
V   003..999  .$3 to $999
V       1000  .Topcode
V      1000+  .$1001+ or more = state median of topcoded
              .values

D  FUELCOST          4            81
        House heating fuel (yearly cost)
V       0000  .N/A (GQ/vacant)
V       0001  .Included in rent or in condo fee
V       0002  .No charge or these fuels not used
V     0003..
        1899  .$3 to $1,899
V       1900  .Topcode
V     1,901+  .$1,901 or more = state median of topcoded
              .value

D  RTAXAMT           2            85
        Property taxes (yearly amount)
V         00  .N/A (GQ/vacant/not owned or being bought/not a
              .one-family house, mobile home or trailer or
              .condo)
V         01  .None
V         02  .$   2 - $  49
V         03  .$  50 - $  99
V         04  .$ 100 - $ 149
V         05  .$ 150 - $ 199
V         06  .$ 200 - $ 249
V         07  .$ 250 - $ 299
V         08  .$ 300 - $ 349
V         09  .$ 350 - $ 399
V         10  .$ 400 - $ 449
```

```
V        11  .$ 450 - $ 499
V        12  .$ 500 - $ 549
V        13  .$ 550 - $ 599
V        14  .$ 600 - $ 649
V        15  .$ 650 - $ 699
V        16  .$ 700 - $ 749
V        17  .$ 750 - $ 799
V        18  .$ 800 - $ 849
V        19  .$ 850 - $ 899
V        20  .$ 900 - $ 949
V        21  .$ 950 - $ 999
V        22  .$1000 - $1099
V        23  .$1100 - $1199
V        24  .$1200 - $1299
V        25  .$1300 - $1399
V        26  .$1400 - $1499
V        27  .$1500 - $1599
V        28  .$1600 - $1699
V        29  .$1700 - $1799
V        30  .$1800 - $1899
V        31  .$1900 - $1999
V        32  .$2000 - $2099
V        33  .$2100 - $2199
V        34  .$2200 - $2299
V        35  .$2300 - $2399
V        36  .$2400 - $2499
V        37  .$2500 - $2599
V        38  .$2600 - $2699
V        39  .$2700 - $2799
V        40  .$2800 - $2899
V        41  .$2900 - $2999
V        42  .$3000 - $3099
V        43  .$3100 - $3199
V        44  .$3200 - $3299
V        45  .$3300 - $3399
V        46  .$3400 - $3499
V        47  .$3500 - $3599
V        48  .$3600 - $3699
V        49  .$3700 - $3799
V        50  .$3800 - $3899
V        51  .$3900 - $3999
V        52  .$4000 - $4099
V        53  .$4100 - $4199
V        54  .$4200 - $4299
V        55  .$4300 - $4399
V        56  .$4400 - $4499
V        57  .$4500 = Topcode
V        58  .$4501 - $54992Ä¿RANGE
V        59  .$5500 - $7499  ³ FOR
V        60  .$7500 or more2ÄÙ  MEDIAN

D  HFILLER2         3            87

D  INSAMT           4            90
        Fire/hazard/flood insurance (yearly amount)
V      0000  .N/A (not owned or being bought/not a one
V            .family house, mobile home, or condo/GQ/vacant)
V      0001  .None
V    0002..
       1299  .$2 to $1,299
V      1300  .Topcode
V     1301+  .$1,301 or more=state median of topcoded values
```

```
D  MORTGAG          1            94
        Mortgage status
V          0  .N/A (not owned or being bought/not a one family
              .house, mobile home, or condo/GQ/vacant)
V          1  .Mortgage deed of trust, or similar debt
V          2  .Contract to purchase
V          3  .None

D  MORTGAG3         5            95
            Mortgage payment (monthly amount)
V      00000  .N/A (not owned or being bought/not a one
              .family house, mobile home, or condo/GQ/vacant)
V      00001  .No regular payment required
V    00002..
       01999  .$2 to $1,999
V      02000  .Topcode
V      02001+ . $2,001 or more = state median of topcoded
              .values

D  TAXINCL          1            100
        Payment include real estate taxes
V          0  .N/A (GQ/vacant/not owned or being bought/
              .not a one family house or condo/not mortgaged/
              .No regular mortgage payment)
V          1  .Yes, taxes included in payment
V          2  .No, taxes paid separately or taxes not required

D  INSINCL          1            101
        Payment include fire/hazard/flood insurance
V          0  .N/A (GQ/vacant/not owned or being bought/
              .Not a one family house, MHT or condo/not
              .mortgaged/no regular mortgage payment)
V          1  .Yes, insurance included in payment
V          2  .No, insurance paid separately or no insurance

D  MORTGAG2         1            102
        Second mortgage or home equity loan status
V          0  .N/A (GQ/vacant/not owned or being bought/
              .not a one family house, mobile home, trailer or
              .condo/not mortgaged/no second mortgage)
V          1  .Yes
V          2  .No

D  MORTAMT2         5            103
        Second mortgage payment (monthly amount)
V       00000 .N/A (GQ/vacant/condo/not owned or being
              .bought/not a one family house/not mortgaged/
              .no second mortgage)
V       00001 .No regular payment required
V     00002..
        00999 .$2 to $999
V       01000 .Topcode
V      01001+ .$1001 or more = state median of topcoded
      .values

D  CONDOFEE         4            108
        Condo fee (monthly amount)
V       0000  .N/A (not owned or being bought/not
V             .condo/GQ/vacant/no condo fee)
V     0001..
        0599  .$1 - $599
V       0600  .Topcode
V      0601+  .$601 or more = state median of topcoded values
```

```
D  MOBLHOME         4            112
        Mobile home costs (yearly amount)
V       0000  .N/A (GQ/vacant/not owned or being bought/
V             .not mobile home/no costs)
V     0001..
V       3399  .$1 - $3,399 (cost in dollars)
V       3400  .Topcode
V      3401+  .$3401 or more = state median of topcoded
V             .values

D  RFARM            1            116
        Farm/nonfarm status
V          0  .N/A (GQ/urban)
V          1  .Rural farm
V              2  .Rural nonfarm

D  RGRENT           4            117
        Gross rent
V       0000  .N/A (GQ/vacant, not rented for cash rent)
V     0001..
        1499  .Gross rent (dollars)
V       1500  .Topcode
V      1501+  .1501 or more = state median of topcoded values

D  RGRAPI           2            121
        Gross rent as a percentage of household income in
         1989
V         00  .N/A (GQ/vacant/not rented for cash rent/owner
              .occupied/no household income)
V         01  . 1% to  9%
V         02  .10% to 14%
V         03  .15% to 19%
V         04  .20% to 24%
V         05  .25% to 29%
V         06  .30% to 34%
V         07  .35% to 39%
V         08  .40% to 49%
V         09  .50% to 59%
V         10  .60% to 69%
V         11  .70% to 79%
V         12  .80% to 89%
V         13  .90% to 99%
V         14  .100% or more

D  HFILLER3         1            123
        Filler

D       ROWNRCST    5            124
        Selected monthly owner costs
V      00000 .N/A (not owned or being bought/not a one
V             .family house, mobile home, or
V             .condo/GQ/vacant/no costs )
V    00001..
       20299  .Monthly owner costs in dollars
V      20300  .Topcode

D  RNSMOCPI         3            129
        Selected monthly owner costs as a percentage of
         household income in 1989
V        000  .N/A (not owned or being bought/not a one family
              .house, mobile home, or condo/GQ/vacant/no HH
              .income)
```

```
V  001..100  .1% to 100%
V       101  .101% or more

D  RRENTUNT         1            132
        Specified rent unit
V         0  .Not specified rent unit
V         1  .Specified rent unit

D  RVALUNT          1            133
        Specified value unit
V         0  .Not specified value unit
V         1  .Specified value unit

D  RFAMINC          7            134
        Family income
V   0000000  .N/A(GQ/vacant/no income)
V -999999..
V   9999999  .Total family income in dollars (See user notes
             .for state maximum and minimum values
             .Includes single person households.)

D  RHHINC           7            141
        Household income
V   0000000  .N/A(GQ/vacant/no income)
V -999999..
    9999999  .Total household income in dollars (See user notes
             .for state maximum and minimum values)

D  RWRKR89          1            148
        Workers in family in 1989
V         0  .N/A (GQ/vacant/non-family household)
V         1  .No workers
V         2  .1 worker
V         3  .2 workers
V         4  .3 or more workers in family

D  RHHLANG          1            149
        Household language
V         0  .N/A (GQ/vacant)
V         1  .English only
V         2  .Spanish
V         3  .Other Indo-European language
V         4  .Asian or Pacific Island language
V         5  .Other language

D  RLINGISO         1            150
        Linguistic isolation
V         0  .N/A (GQ/vacant)
V         1  .Not linguistically isolated
V         2  .Linguistically isolated

D  RHHFAMTP         2            151
        Household/family type
V        00  .N/A (GQ/vacant)
V        01  .Married-couple family household
V             Other family household:
V        02  .Male householder
V        03  .Female householder
V             Nonfamily household:
V             Male householder:
V        11  .Living alone
V        12  .Not living alone
V             Female householder:
```

```
V        21  .Living alone
V        22  .Not living alone

D  RNATADPT        2            153
        Number of own natural born/adopted children in
         household (unweighted)
V        00  .N/A(GQ/vacant/no own natural born/adopted
V            .children)
V    01..28  .Number of own children natural born/adopted
             .children in household

D  RSTPCHLD        2            155
        Number of own stepchildren in household (unweighted)
V        00  .N/A(GQ/vacant/no own stepchildren)
V    01..28  .Number of own stepchildren in household

D  RFAMPERS        2            157
        Number of persons in family (unweighted)
V        00  .N/A (GQ/vacant/non-family household)
V    01..29  .Number of persons in family

D  RNRLCHLD*       2            159
        Number of related children in household (unweighted)
V        00  .N/A (GQ/vacant/no related children)
V    01..28  .Number of related children in household

D  RNONREL         1            161
            Presence of nonrelatives in household
V         0  .N/A (No nonrelatives in household/GQ/vacant)
V         1  .1 or more nonrelatives in household

D  R18UNDR         1            162
        Presence of person under 18 years in household
V         0  .N/A (No person under 18 in household/GQ/vacant)
V         1  .1 or more person under 18 in household

D  R60OVER         1            163
        Presence of persons 60 years and over in household
V         0  .N/A (No person 60 and over/GQ/vacant)
V         1  .1 person 60 and over (unweighted)
V         2  .2 or more person 60 and over (unweighted)

D  R65OVER         1            164
        Presence of person 65 years and over in household
V         0  .N/A (No person 65 and over/GQ/vacant)
V         1  .1 person 65 and over (unweighted)
V         2  .2 or more person 65 and over (unweighted)

D  RSUBFAM         1            165
        Presence of subfamilies in Household
V         0  .N/A (No subfamilies or not
V            .applicable/GQ/vacant)
V         1  .1 or more subfamilies

D  AUNITS1         1            166
        Units in structure allocation
V         0  .No
V         1  .Yes

D  AROOMS          1            167
        Rooms allocation
V         0  .No
V         1  .Yes
```

```
D  ATENURE          1            168
        Tenure allocation
V         0  .No
V         1  .Yes

D  AACRES10         1            169
        On ten acres or more allocation
V         0  .No
V         1  .Yes

D  ACOMMUSE         1            170
        Business or medical office on property allocation
V         0  .No
V         1  .Yes

D  AVALUE           1            171
        Value allocation
V         0  .No
V         1  .Yes

D  ARENT1           1            172
        Monthly rent allocation
V         0  .No
V         1  .Yes

D  AMEALS           1            173
        Meals included in rent allocation
V         0  .No
V         1  .Yes

D  AVACNCY2         1            174
        Vacancy status allocation
V             0  .No
V         1  .Yes

D  AVACNCY3         1            175
        Boarded up status allocation
V         0  .No
V         1  .Yes

D  AVACNCY4         1            176
        Months vacant allocation
V         0  .No
V         1  .Yes

D  AYRMOVED         1            177
        When moved into this house or apartment allocation
V         0  .No
V         1  .Yes

D  ABEDROOM         1            178
        Number of bedrooms allocation
V         0  .No
V         1  .Yes

D  APLUMBNG         1            179
        Complete plumbing facilities allocation
V         0  .No
V         1  .Yes

D  AKITCHEN         1            180
        Complete kitchen facilities allocation
```

```
V         0  .No
V         1  .Yes

D  APHONE          1             181
        Telephones in house allocation
V         0  .No
V         1  .Yes

D  AVEHICLE        1             182
        Vehicles available by household allocation
V         0  .No
V         1  .Yes

D  AFUEL           1             183
        House heating fuel allocation
V         0  .No
V         1  .Yes

D  AWATER          1             184
        Source of water allocation
V         0  .No
V         1  .Yes

D  ASEWER          1             185
        Sewage disposal allocation
V         0  .No
V         1  .Yes

D  AYRBUILT        1             186
        When structure first built allocation
V         0  .No
V         1  .Yes from not answered
V         2  .Yes “don’t know”

D  ACONDO          1             187
        House or apartment part of condominium allocation
V         0  .No
V         1  .Yes

D  AONEACRE        1             188
        House on less than 1 acre allocation
V         0  .No
V         1  .Yes

D  AAGSALES        1             189
        1989 Sales of Agricultural Products allocation
V         0  .No
V         1  .Yes

D  AELECCST        1             190
        Electricity (yearly cost) allocation
V         0  .No
V         1  .Yes


D  AGASCST         1             191
        Gas (yearly cost) allocation
V         0  .No
V         1  .Yes

D  AWATRCST        1             192
        Water (yearly cost) allocation
V         0  .No
```

```
V         1  .Yes

D  AFUELCST*         1             193
        House heating fuel (yearly cost) allocation
V         0  .No
V         1  .Yes

D  ATAXAMT           1             194
        Taxes on property allocation
V         0  .No
V         1  .Yes

D  AINSAMT           1             195
        Fire, hazard, flood insurance allocation
V         0  .No
V         1  .Yes

D  AMORTG            1             196
        Mortgage status allocation
V       0 .No
V       1 .Yes no answer
V       2 .Yes from junior mortgage

D  AMORTG3           1             197
        Regular mortgage payment allocation
V       0 .No
V       1 .Yes

D  ATAXINCL          1             198
        Payment include real estate taxes allocation
V       0 .No
V       1 .Yes

D  AINSINCL          1             199
        Payment include fire, hazard, flood insurance
         allocation
V       0 .No
V       1 .Yes

D  AMORTG2           1             200
        Second mortgage status allocation
V       0 .No
V       1 .Yes

D  AMRTAMT2          1             201
            Second mortgage payment allocation
V       0 .NO
V       1 .Yes

D  ACNDOFEE          1             202
        Condominium fee allocation
V       0 .No
V       1 .Yes

D  AMOBLHME          1             203
        Mobile home costs allocation
V       0 .No
V       1 .Yes

D  FILLER            28            204
```

Following is a list of the demographic options that are available for persons and the allowed values for each demographic item.

```
DATA                    SIZE     BEGIN

D  RECTYPE          1            1
        Record Type
V          P .Person Record

D  SERIALNO         7            2
V  0000000..
V    9999999 .Housing unit/GQ person serial number unique
V            .identifier assigned within state or state group

D  RELAT1           2            9
        Relationship
V         00 .Householder
V         01 .Husband/wife
V         02 .Son/daughter
V         03 .Stepson/stepdaughter
V         04 .Brother/sister
V         05 .Father/mother
V         06 .Grandchild
V         07 .Other relative
              Not related
V         08 .Roomer/boarder/foster child
V         09 .Housemate/roommate
V         10 .Unmarried partner
V         11 .Other nonrelative
              Group quarters
V         12 .Institutionalized person
V         13 .Other persons in group quarters

D  SEX              1           11
        Sex
V          0 .Male
V          1 .Female

D  RACE             3           12
        Recoded detailed race code (Appendix C)
V    001-037 .(See appendix C)
V    301-327 .American Indian Tribes

D  AGE              2           15
        Age
V         00 .Less than 1 year
V     01..89 .Age in years
V         90 .90 or more years old

D  MARITAL          1           17
        Marital status
V          0 .Now married, except separated
V          1 .Widowed
V          2 .Divorced
V          3 .Separated
V          4 .Never married or under 15 years old

D  PWGT1            4            18
        Person's weight
V     0001..
V       1152 .Person's weight

D  PFILLER1         4            22
        Filler
```

```
D  REMPLPAR         3             26
        Employment status of parents
V        000 .N/A (not own child of householder, and not
             .child in subfamily)
V           Living with two parents:
V             Both parents in labor force:
V       111     .Both parents at work 35 or more hours
V       112     .Father only at work 35 or more hours
V       113     .Mother only at work 35 or more hours
V       114     .Neither parent at work 35 or more hours
V             Father only in labor force:
V       121     .Father at work 35 or more hours
V       122     .Father not at work 35 or more hours
V             Mother only in labor force:
V       133     .Mother at work 35 or more hours
V       134     .Mother not at work 35 or more hours
V       141   Neither parent in labor force
V           Living with one parent:
V             Living with father:
V       211     .Father at work 35 or more hours
V       212     .Father not at work 35 or more hours
V       213     .Father not in labor force
V             Living with mother:
V       221     .Mother at work 35 or more hours
V       222     .Mother not at work 35 or more hours
V       223     .Mother not in labor force

D  RPOB             2            29
        Place of birth (Recode)
V       10   .Born in State of residence
V           Born in other State in the U.S.:
V       21   .Northeast
V       22   .Midwest
V       23   .South
V       24   .West
V           U.S. outlying areas:
V       31   .Puerto Rico
V       32   .American Samoa
V       33   .Guam
V       34   .Northern Marianas
V       35   .US Virgin Islands
V       36   .Elsewhere
V       40   .Born abroad of American parents
V           Foreign born:
V       51   .Naturalized citizen
V       52   .Not a citizen

D  RSPOUSE          1             31
        Married, spouse present/spouse absent
V        0   .N/A (less than 15 years old)
V        1   .Now married, spouse present
V        2   .Now married, spouse absent
V        3   .Widowed
V        4   .Divorced
V        5   .Separated
V        6   .Never married

D  ROWNCHLD         1             32
            *Own child (see Appendix B, page 14)
V       * 1   .Own child
V       * 0   .Not own child

D  RAGECHLD         1             33
```

```
        Presence and age of own children
V        * 0 .N/A (male)
V          1 .With own children under 6 years only
V          2 .With own children 6 to 17 years only
V          3 .With own children under 6 years and 6 to 17
             .years
V        *  4 .No own children (.incl. females under 16 years)
D  RRELCHLD         1            34

            *Related child (see Appendix B, Page 14)
V         * 1 .Related child
V         * 0 .Not related child

D  RELAT2           1            35
        Detailed relationship (other relative)
V          0 .N/A (GQ/not other relative)
V          1 .Son-in-law/daughter-in-law
V          2 .Father-in-law/mother-in-law
V          3 .Brother-in-law/sister-in-law
V          4 .Nephew/niece
V          5 .Grandparent
V          6 .Uncle/aunt
V          7 .Cousin
V          8 .Other related by blood or marriage
V          9 .Other relative

D  SUBFAM2          1            36
        Subfamily number
V          0 .N/A (GQ/not in a subfamily)
V          1 .In subfamily 1
V          2 .In subfamily 2
V          3 .In subfamily 3

D  SUBFAM1          1            37
        Subfamily relationship
V          0 .N/A (GQ/not in a subfamily)
V          1 .Husband/wife
V          2 .Parent in a parent/child subfamily
V          3 .Child in subfamily

D  HISPANIC         3            38
        Detailed Hispanic origin code (See appendix I)
V  000,006.. .
         199 . Not hispanic
V  001,210..
         220 .Mexican, mex-am
V  002,261..
         270 .Puerto Rican
V  003,271..
         274 .Cuban
V   221..230 .Central American
V   231..249 .South American
V   275..289 .Dominican
V  004,200..
   209,250..
         260
V   290..401 .Other Hispanic

D  POVERTY          3            41
        Person poverty status recode (See appendix B)
V        000 .N/A
V   001..500 .% Below or above poverty status value
V        501 .501% or more of poverty value
```

```
D  POB              3            44
        Place of birth (Appendix I)
V 001..056   .FIPS State code (U.S. States and D.C.)
V 060..099   .Puerto Rico (072) or U.S. outlying area
V 100..553   .Foreign country
V      554   .At sea
V      555   .Abroad, not specified

D  CITIZEN          1            47
        Citizenship
V        0   .Born in the U.S.
V        1   .Born in Puerto Rico, Guam, and outlying areas
V        2   .Born abroad of American parents
V        3   .U.S. citizen by naturalization
V        4   .Not a citizen of the U.S.

D  IMMIGR           2            48
        Year of entry
V       00   .Born in the U.S.
V       01   .1987 to 1990
V       02   .1985 to 1986
V       03   .1982 to 1984
V       04   .1980 or 1981
V       05   .1975 to 1979
V       06   .1970 to 1974
V       07   .1965 to 1969
V       08   .1960 to 1964
V       09   .1950 to 1959
V       10   .Before 1950

D  SCHOOL           1            50
        School enrollment
V        0   .N/A (less than 3 years old)
V        1   .Not attending school
V        2   .Yes, public school, public college
V        3   .Yes, private school, private college

D  YEARSCH          2            51
        Educational attainment
V       00 .N/A (less than 3 years old)
V       01 .No school completed
V       02 .Nursery school
V       03 .Kindergarten
V       04 .1st, 2nd, 3rd, or 4th grade
V       05 .5th, 6th, 7th, or 8th grade
V       06 .9th grade
V       07 .10th grade
V       08 .11th grade
V       09 .12th grade, no diploma
V       10 .High school graduate, diploma or GED
V       11 .Some college, but no degree
V       12 .Associate degree in college, occupational program
V       13 .Associate degree in college, academic program
V       14 .Bachelor's degree
V       15 .Master's degree
V       16 .Professional degree
V       17 .Doctorate degree

D  ANCSTRY1         3            53
        Ancestry – first entry (See appendix I)
V 001..998   .Ancestry codes – first entry
V      999   .Not reported
```

```
D  ANCSTRY2        3            56
     Ancestry – second entry (See appendix I)
V       000 .No secondary ancestry
V   001..998 .Ancestry codes
V       999 .Not reported

D  MOBILITY        1            59
     Mobility status (lived here on April 1, 1985)
V          0 .N/A(less than 5 years old)
V          1 .Yes same house (nonmovers)
V          2 .No, different house (movers)

D  MIGSTATE        2            60
     Migration – State or foreign country code
      (Appendix I)
V         00 .N/A (person less than 5 years old/lived
V             .in same house in 1985)
V     01..56 .FIPS state code (U.S. States and D.C.)
V         72 .Puerto Rico
V         98 .Other abroad in 1985
V         99 .State not identified (B sample)

D  MIGPUMA         5            62
     Migration PUMA (state dependent)
V      00000 .N/A (person less than 5 years old/lived in
             .same house in 1985)
V    00100..
       99800 .Migration PUMA  (Not coded to tract level)
V      99900 .Abroad

D  LANG1           1            67
     Language other than English at home
V          0 .N/A (less than 5 years old)
V          1 .Yes, speaks another language
V          2 .No, speaks only English

D  LANG2           3            68
     Language spoken at home (See appendix I)
V   000..600 .N/A (less than 5 years old/speaks only
             .English)
V   601..999 .Specific language codes

D  ENGLISH         1            71
     Ability to speak English
V          0 .N/A (less than 5 years old/speaks only English)
V          1 .Very well
V          2 .Well
V          3 .Not well
V          4 .Not at all

D  MILITARY        1            72
     Military service
V          0 .N/A (less than 16 years old)
V          1 .Yes, now on active duty
V          2 .Yes, on active duty in past, but not now
V          3 .Yes, service in reserves or national guard only
V          4 .No service

D  RVETSERV        2            73
     Veteran period of service
V       00   .N/A (less than 16 years old, no active duty)
V       01   .September 1980 or later only
```

```
V       02   .May 1975 to August 1980 only
V       03   .May 1975 to August 1980 and September 1980
V            .or later only
V       04   .Vietnam era, no Korean conflict, no WWII
V       05   .Vietnam era and Korean conflict, no WWII
V       06   .Vietnam era and Korean conflict and WWII
V       07   .February 1955 to July 1964 only
V       08   .Korean conflict, no Vietnam era, no WWII
V       09   .Korean conflict and WWII, no Vietnam era
V       10   .WWII, no Korean conflict, no Vietnam era
V       11   .Other service

D  SEPT80           1            75
         Served September 1980 or later
V        0   .(Did not serve this period/less than 16 years
V            .old)
V        1   .Served this period

D  MAY75880         1            76
        Served May 1975 to August 1980
V        0   .(Did not serve this period/less than 16 years
V            .old)
V        1   .Served this period

D  VIETNAM          1            77
        Served Vietnam era (August 1964 – April 1975)
V        0   .(Did not serve this period/less than 16 years
V            . old)
V        1   .Served this period

D  FEB55            1            78
        Served February 1955 – July 1964
V        0   .(Did not serve this period/less than 16 years
V            .old)
V        1   .Served this period

D  KOREAN           1            79
        Served Korean conflict (June 1950 – January 1955)
V        0   .(Did not serve this period/less than 16 years
V            .old)
V        1   .Served this period

D  WWII             1            80
        Served World War II (September 1940 – July 1947)
V        0   .(Did not serve this period/less than 16 years
V            .old)
V        1   .Served this period

D  PFILLER2         1            81
        Filler

D  OTHRSERV         1            82
        Served any other time
V        0   .(Did not serve this period/less than 16 years
V            .old)
V        1   .Served this period

D  YRSSERV          2            83
        Years of active duty military service
V       00   .N/A (less than 16 years/no active duty military
V            .service)
V       01   .1 Year or less of service
V   02..49   .2 to 49 years of service
```

```
V       50   .50 or more years of service

D  DISABL1          1            85
        Work limitation status
V          0 .N/A (less than 16 years, and selected persons in
V            .GQs – See appendix C)
V          1 .Yes, limited in kind or amount of work
V          2 .No, not limited

D  DISABL2          1            86
        Work prevented status
V          0 .N/A(less than 16 years, and selected persons in
V            .GQs – See appendix C)
V          1 .Yes, prevented from working
V          2 .No, not prevented from working

D  MOBILLIM         1            87
        Mobility limitation
V          0 .N/A (less than 15 years/institutionalized
V                .person, and selected persons in GQs –
V            .See appendix C)
V          1 .Yes, has a mobility limitation
V          2 .No, does not have a mobility limitation

D  PERSCARE         1            88
        Personal care limitation
V          0 .N/A (less than 15 years/institutionalized
V            .person, and selected persons in GQs –
V            .See appendix C)
V          1 .Yes, has a personal care limitation
V          2 .No, does not have a personal care limitation

D  FERTIL           2            89
        Number of children ever born
V         00 .N/A (less than 15 years/male)
V         01 .No children
V         02 .1 Child
V         03 .2 Children
V         04 .3 Children
V         05 .4 Children
V         06 .5 Children
V         07 .6 Children
V         08 .7 Children
V         09 .8 Children
V         10 .9 Children
V         11 .10 Children
V         12 .11 Children
V         13 .12 or more children

D  RLABOR           1            91
        Employment status recode
V          0 .N/A (less than 16 years old)
V          1 .Civilian employed, at work
V          2 .Civilian employed, with a job but not at work
V          3 .Unemployed
V          4 .Armed forces, at work
V          5 .Armed forces, with a job but not at work
V          6 .Not in labor force

D  WORKLWK          1            92
        Worked last week
V          0 .N/A (less than 16 years old/not at work/
V            .unemployed/NILF/Q21A not reported)
```

```
V          1 .Worked
V          2 .Did not work

D  HOURS            2            93
        Hours worked last week
V       00   .N/A (less than 16 years old/not at
V            .work/unemployed/NILF)
V    01..98  .1 to 98 hours worked last week
V       99   .99 or more hours worked last week

D  POWSTATE         2            95
        Place of work – state – (Appendix I)
V       00   .N/A (not a worker—not in the labor force,
V            .including persons under 16 years; unemployed;
V            .employed, with a job not at work; Armed Forces,
             .With a job but not at work)
V     01-56  .FIPS state code (U.S. States and D.C.)
V       98   .Abroad
V       99   .State not identified

D  POWPUMA          5            97
        Place of work PUMA (State dependent)
V     00000  .N/A (not a worker—not in the labor force,
V            .including persons under 16 years;
V            .unemployed; employed, with a job but not at
V            .work; Armed Forces, with a job but not at
V            .work)
V  00100..
      99800  .PUMA of work (Not coded to tract level)
V     99900  .Abroad

D  MEANS            2            102
        Means of transportation to work
V       00   .N/A (not a worker—not in the labor force,
V            .including persons under 16 years;  unemployed;
V            .employed, with a job but not at work;  Armed
V            .Forces, with a job but not at work)
V       01   .Car, truck, or van
V       02   .Bus or trolley bus
V       03   .Streetcar or trolley car
V       04   .Subway or elevated
V       05   .Railroad
V       06   .Ferryboat
V       07   .Taxicab
V       08   .Motorcycle
V       09   .Bicycle
V       10   .Walked
V       11   .Worked at home
V       12   .Other method

D  RIDERS           1            104
        Vehicle occupancy
V        0   .N/A (not a worker or worker whose means of
V            .transportation to work was not car, truck,
V            .or van)
V        1   .Drove alone
V        2   .2 People
V        3   .3 People
V        4   .4 People
V        5   .5 People
V        6   .6 People
V        7   .7 to 9 people
V        8   .10 or more people
```

```
D  DEPART          4            105
        Time of departure for work - hour and minute
V       0000 .N/A (not a worker or worker who worked at
             .home)
V     0001..
        2400 .Time (hour and minute of departure for
             .work) (2400 midnight)

D  TRAVTIME        2            109
        Travel time to work
V         00 .N/A (not a worker or worker who worked at home)
V     01..98 .1 to 98 minutes to get to work
V         99 .99 Minutes or more to get to work

D  TMPABSNT        1            111
        Temporary absence from work
V          0 .N/A (less than 16 years old/at work/did not
             .report Q25)
V          1 .Yes, on layoff
V          2 .Yes, on vacation, temporary illness, labor
             .dispute
V          3 .No

D  LOOKING         1            112
        Looking for work
V          0 .N/A (less than 16 years old/at work/did not
             .report Q26A)
V          1 .Yes
V          2 .No

D  AVAIL           1            113
        Available for work
V          0 .N/A (less than 16 years/at work/not looking/
             .Q26A = 0/did not report Q26B)
V          1 .No, already has a job
V          2 .No, temporarily ill
V          3 .No, other reasons (in school, etc.)
V          4 .Yes, could have taken a job

D  YEARWRK         1            114
        Year last worked
V          0 .N/A (less than 16 years old)
V          1 .1990
V          2 .1989
V          3 .1988
V          4 .1985 to 1987
V          5 .1980 to 1984
V          6 .1979 or earlier
V          7  .Never worked

D  INDUSTRY        3            115
        Industry
V        000 .N/A (less than 16 years old/unemployed who
             .never worked/nilf who last worked prior to
             .1985)
V 010..992   .specific industry codes (see appendix I)

D  OCCUP           3            118
        Occupation
V      000   .N/A (less than 16 years old/unemployed who
             .never worked/nilf who last worked prior to
             .1985)
```

```
V 003..909   .specific occupation codes (see appendix I)

D  CLASS            1            121
        Class of worker
V        0   .N/A (less than 16 years old/unemployed who
V            .never worked/NILF who last worked prior to
V            .1985)
V        1   .employee of a private for profit company or
V            .business or of an individual, for wages,
V            .salary, or commissions
V        2   .Employee of a private not-for-profit,
V            .tax-exempt, or charitable organization
V        3   .Local government employee (city, county, etc.)
V        4   .State government employee
V        5   .Federal government employee
V        6   .Self-employed in own not incorporated
V            .business, professional practice, or farm
V        7   .Self-employed in own incorporated
V            .business, professional practice or farm
V        8   .Working without pay in family business or farm
V        9   .Unemployed, last worked in 1984 or earlier

D  WORK89           1            122
        Worked last year (1989)
V        0   .N/A (less than 16 years old)
V        1   .Worked last year
V        2   .Did not work last year

D  WEEK89           2            123
        Weeks worked last year (1989)
V       00   .N/A (less than 16 years old/did not work in
             .1989)
V   01..52   .1 to 52 weeks worked last year

D  HOUR89           2            125
        Usual hours worked per week last year (1989)
V       00   .N/A (less than 16 years old/did not work in
                 .1989)
V   01..98   .1 To 98 usual hours
V       99   .99 Or more usual hours

D  REARNING         6            127
        Total person’s earnings
V     000000 .N/A (no earnings)
V     -19996 .Loss of $19996 or more
V   -19995..
      283999 .Total person’s earnings in dollars
V     284000 .$284000 = Topcode
V    284001+ .State medians included

D  RPINCOME         6            133
        Total person’s income (signed)
V    000000   .N/A (no income)
V    -29997   .Loss of $29997 or more
V  -29996..
     400999  .Total person’s income in dollars
V    401000  .Topcode of total person’s income
V   401001+  .State medians included

D  INCOME1          6            139
        Wages or salary income in 1989
V    000000  .N/A (less than 16 years old/none)
V  000001..
```

```
V    139999  .$1 – 139,999
V    140000  .Topcode
V   140001+  .140001 or more = state median of topcoded
V            .values

D  INCOME2          6            145
            Nonfarm self-employment income in 1989 (signed)
V   000000   .N/A (less than 16 years/none)
V   -09999   .Loss of $9,999 or more
V   -00001..
V     -09998 .Loss $1 to $9,998
V     000001 .Break even or $1
V   000002..
      089999 .$2 To $89999
V     090000 .Topcode
V    090001+ .$90,001 or more = state median of topcoded
             .values

D  INCOME3          6            151
        Farm self-employment income in 1989 (signed)
V     000000 .N/A (less than 16 years/none)
V     -09999 .Loss of $9,999 or more
V  -00001 to
      -09998 .Loss $1 to $9,998
V          1 .Break even or $1
V   000002..
      053999 .$2 To $53999
V     054000 .Topcode
V    054001+ .$54001 or more = state median of
             .topcoded values

D  INCOME4          6            157
        Interest, dividends, and net rental income in 1989 (signed)
V     000000 .N/A (less than 15 years/none)
V     -09999 .Loss of $9,999 or more
V  -00001 to
      -09998 .Loss $1 to $9,998
V          1 .Break even or $1
V   000002..
      039999 .$2 To $39999
V     040000 .Topcode
V    040001+ .$40001 or more = state median of
             .topcoded values
D  INCOME5          5            163
        Social security income in 1989
V      00000 .N/A (less than 15 years/none)
V    00001..
       16999 .$1 to $16999
V      17000 .Topcode
V     17001+ .17001 or more = state median of topcoded
             .values

D  INCOME6          5            168
        Public assistance income in 1989
V      00000 .N/A (less than 15 years/none)
V    00001..
        9999 .$1 To $9999
V      10000 .Topcode
V     10001+ .$10001 or more = state median

D  INCOME7          5            173
        Retirement income in 1989
V    00000   .N/A (less than 15 years/none)
```

```
V    00001..
       29999 .$1 to $29999
V      30000 .Topcode
V     30001+ .$30001 or more = state median of topcoded
             .values

D  INCOME8          5            178
        All other income in 1989
V    00000   .N/A (less than 15 years/none)
V  00001..
     19999   .$1 to $19999
V    20000   .Topcode
V   20001+   .$20,001 or more = state median of topcoded
             .values

D  AAUGMENT         1            183
        Augmented person (see text pp. C-5)
V        0   .No
V        1   .Yes

D  ARELAT1          1            184
        Relationship allocation flag
V        0   .No
V        1   .Yes

D  ASEX             1            185
        Sex allocation flag
V        0   .No
V        1   .Yes

D  ARACE            1            186
        Detailed race allocation flag
V        0   .No
V        1   .Yes

D  AAGE              1           187
        Age allocation flag
V        0   .No
V        1   .Yes

D  AMARITAL         1            188
        Marital status allocation flag
V        0   .No
V        1   .Yes

D  AHISPAN          1            189
        Detailed Hispanic origin allocation flag
V        0   .No
V        1   .Yes
D  ABIRTHPL         1            190
        Place of birth
V        0   .No
V        1   .Yes

D  ACITIZEN         1            191
        Citizenship allocation flag
V        0   .No
V        1   .Yes

D  AIMMIGR          1            192
        Year of entry allocation flag
V       0 .No
V       1 .Yes
```

```
D  ASCHOOL          1            193
     School enrollment allocation flag
V       0 .No
V       1 .Yes

D  AYEARSCH         1            194
     Highest education allocation flag
V          0 .No
V          1 .Yes

D  AANCSTR1         1            195
     First ancestry allocation flag
V          0 .No
V          1 .Yes

D  AANCSTR2         1            196
     Second ancestry allocation flag
V          0 .No
V          1 .Yes

D  AMOBLTY          1            197
     Mobility status allocation flag
V          0 .No
V          1 .Yes

D  AMIGSTATE        1            198
     Migration state allocation flag
V          0 .No
V          1 .Yes

D  ALANG1           1            199
     Language other than English allocation flag
V          0 .No
V          1 .Yes

D  ALANG2           1            200
     Language spoken at home allocation flag
V          0 .No
V          1 .Yes

D  AENGLISH         1            201
     Ability to speak English allocation flag
V          0 .No
V          1 .Yes

D  AVETS1           1            202
     Military service allocation flag
V          0 .No
V          1 .Yes

D  ASERVPER         1            203
     Military periods of service allocation flag
V          0 .No
V          1 .Yes
D  AYRSSERV         1            204
     Years of military service allocation flag
V          0 .No
V          1 .Yes

D  ADISABL1         1            205
     Work limitation status allocation flag
V        0   .No
```

```
V       1    .Yes

D  ADISABL2         1            206
        Work prevention status allocation flag
V       0    .No
V       1    .Yes

D  AMOBLLIM         1            207
        Mobility limitation status allocation flag
V       0    .No
V       1    .Yes

D  APERCARE         1            208
        Personal care limitation status allocation flag
V       0    .No
V       1    .Yes

D  AFERTIL          1            209
        Children ever born allocation flag
V       0    .No
V       1    .Yes

D  ALABOR           1            210
        Employment status recode allocation flag
V       0    .No
V       1    .Yes

D  AHOURS           1            211
        Hours worked last week allocation flag
V       0    .No
V       1    .Yes

D  APOWST           1            212
        Place of work state allocation flag
V       0    .No
V       1    .Yes

D  AMEANS           1            213
        Means of transportation to work allocation flag
V       0    .No
V       1    .Yes

D  ARIDERS          1            214
        Vehicle occupancy allocation flag
V       0    .No
V       1    .Yes

D  ADEPART          1            215
        Time of departure to work allocation flag
V       0    .No
V       1    .Yes

D  ATRAVTME         1            216
        Travel time to work allocation flag
V       0    .No
V       1    .Yes

D  ALSTWRK          1            217
        Year last worked allocation flag
V       0    .No
V       1    .Yes

D  AINDUSTR         1            218
```

```
        Industry allocation flag
V       0 .No
V       1 .Yes

D  AOCCUP          1            219
        Occupation allocation flag
V       0 .No
V       1 .Yes

D  ACLASS          1            220
        Class of worker allocation flag
V          0 .No
V          1 .Yes

D  AWORK89         1            221
        Worked last year allocation flag
V          0 .No
V          1 .Yes

D  AWKS89          1            222
        Weeks worked in 1989 allocation flag
V          0 .No
V          1 .Yes

D  AHOUR89         1            223
        Usual hours worked per week in 1989 allocation flag
V          0 .No
V          1 .Yes

D  AINCOME1         1           224
        Wages and salary income allocation flag
V          0 .No
V          1 .No (derived)
V          2 .Yes

D  AINCOME2        1            225
        Nonfarm self-employment income allocation flag
V          0 .No
V          1 .No (derived)
V          2 .Yes

D  AINCOME3        1            226
        Farm self-employment income allocation flag
V          0 .No
V          1 .No (derived)
V          2 .Yes

D  AINCOME4        1            227
        Interest, dividend, and net rental income allocation flag
V          0 .No
V          1 .No (derived)
V          2 .Yes

D  AINCOME5        1            228
        Social security income allocation flag
V          0 .No
V          1 .No (derived)
V          2 .Yes
D  AINCOME6        1            229
        Public assistance allocation flag
V        0   .No
V        1   .No (derived)
V        2   .Yes
```

```
D  AINCOME7         1             230
        Retirement income allocation flag
V        0   .No
V            1   .No (derived)
V        2   .Yes

D  AINCOME8         1             231
        All other income allocation flag
V        0   .No
V        1   .No (derived)
V        2   .Yes
```